Final Report Bachelor's Thesis

P&O Entrepreneurship on behalf of Relu BV

H01Q3C

Anna Proost (r0850180)
Arne Van Damme (r0856420)
Astrid Dewever (r0855007)
Dag Malstaf (r0799028)
Jonas De Schouwer (r0861232)
Kjell Mesotten (r0848158)
Lars Depuydt (r0849855)
Senne Vanden Eynde (r0853326)
Thomas Speltincx (r0856598)

May 9, 2023
Academic year **2022–2023**

Originality Declaration

We declare that we are the original and sole authors of this submitted paper, except for feedback and support provided to us by the educational team responsible for the guidance, and except for authorised collaboration agreed upon with the same educational team. With regard to this paper, we also declare that:

1. We have taken note of the text on academic integrity at `https://eng.kuleuven.be/en/study/plagiarism`.

2. No plagiarism has been committed as described at `https://eng.kuleuven.be/en/study/plagiarism#Definitie:%20wat%20is%20plagiaat?`.

3. All experiments, tests, and measurements have been carried out as described in this present text, and no data or measurements have been manipulated.

4. All source references and citations have been included in good faith.

Contents

List of Figures

1 Introduction — Astrid Dewever

The dental industry is at a crossroads. For years, advancements in technology have played a significant role in shaping the field, from the introduction of dental implants to the use of 3D printing in creating dental prostheses. Yet now, with the advent of virtual reality (VR) and the metaverse, the landscape of dentistry is poised for another seismic shift. Looking towards the future, one thing is becoming increasingly clear:
the future of oral medicine is not in the hands of dentists but in the code of software engineers.
In this paper, the development of a tool is explored that uses VR to move jaws and plan maxillofacial surgery.

The problem is that the current workflow for oral surgery procedures is inefficient, costly, and unsustainable, leading to a suboptimal patient experience and decreased profitability for dental practices. In the following section, the project is situated in the proper context and it is made clear that the current process requires the creation of physical models and multiple redundant steps in the pipeline.

When Relu, a company specialised in software for teledentistry, commissioned a project to improve the workflow for oral surgery procedures, a team of nine computer science students was assembled to tackle the challenge. In order to develop the optimal tool, a rigorous development process was followed. This process began with workshops in Agile Development and brainstorming sessions to determine the most effective approach. Four minimum viable products (MVPs) were created, each of which proposed a different approach to addressing the challenges associated with maxillofacial surgery planning. The MVPs were then presented to the main stakeholders, including Relu and the professors involved in the project. After receiving feedback from these stakeholders, one MVP was selected for further development throughout the rest of the academic year in order to push the boundaries of what is possible with modern technology. The result is an innovative solution that utilises virtual reality to revolutionise the way dental professionals plan maxillofacial surgeries within a digital environment.

This paper will discuss the development, implementation, and evaluation of a web application with a virtual reality environment designed to improve the workflow for oral surgery procedures in terms of efficiency, cost-effectiveness, and sustainability.

To be exact, the paper is structured as follows. The second chapter introduces the research problem and presents the research questions and objectives. Chapter 3 presents the methodology used in this study, including descriptions of the workshops conducted (such as constructive feedback, brainstorming, Lean canvas business model, pitching workshop by KICK, and Agile development) and how the team was divided to accomplish the research objectives. Chapter 4 provides an overview of the technologies and tools utilised in the implementation of the proposed solution. The chapter includes detailed discussions on GoLang, Docker, Google Cloud and Firebase, Next.js, Three.js and Cannon.js, which were used to develop the system. Chapter 5 of the paper focuses on presenting and analyzing the implementation of the research project. After first giving an overview, the chapter is structured into several sections, each discussing a specific aspect of the project, such as the back end and front end functionality, the Virtual Reality (VR) approach and specifications, and the collision handling and jaw movement in VR. Chapter 6 presents and discusses the results obtained from the web app and VR environment, which meets the research questions and objectives, as described in the second chapter. Chapter 7 provides a comprehensive analysis of the project through the use of a SWOT framework. The chapter highlights the strengths, weaknesses, opportunities, and threats associated with the project, offering a detailed overview of the project's internal and external environment. Chapter 8 of this thesis provides an in-depth business analysis of the project, inspired by the Lean Canvas framework, including unique value propositions, customer segments, revenue streams, cost structure, key metrics, and unfair advantage. This chapter aims to present an exhaustive overview of the business aspects of the project and to provide valuable insights for potential investors and involved stakeholders. Chapter 9 delves into the perspectives and feedback of the two stakeholder groups involved in the project, analyzing their influence on the project's development and outcomes. It highlights the ways in which their opinions were incorporated and the resulting impact on the project. Chapter 10 examines the potential for future growth and expansion of the project, through an in-depth analysis of scalability options such as horizontal and vertical scaling, and the utilization of cloud computing technologies. Chapter 11 proposes future prospects for the project, including potential features to be added and suggestions for further development. In Chapter 12, the final destination is reached and the journey comes to a poetic end. The project's purpose is fulfilled and the conclusion offers a reflective, insightful, and meaningful farewell to the readers.

2 Situating the project — Jonas De Schouwer

For this P&O Entrepreneurship project, the team had the opportunity to collaborate with Relu, a Leuven-based start-up focused on creating support software for the medical industry, dentistry in particular. Their primary product is the *Virtual Patient Creator*, a web application that can automatically segment medical 3D scans using artificial intelligence.

The case Relu presented revolves around the treatment process for oral surgery operations. Currently, this process goes as follows:

1. A patient has a problem and goes to the dentist.
2. The dentist takes a physical negative impression or intraoral digital scan of the patient's teeth.
3. A physical 3D model is made by filling the negative impression with plaster, or by printing the digital scan with a 3D printer.
4. The dental surgeon studies the problem and uses the physical model to find the optimal fit between the patient's upper and lower jaw. This result determines which surgery will be required and the surgeon locks the model in this desired target position.
5. The locked model is scanned again to generate a digital version of the optimal fit.
6. A biomedical engineer designs a mould that the patient can bite on during surgery, which ensures the proper dental occlusion. Once the patient assumes this position, the oral surgeon can begin the operation.

However, this workflow is both costly and inefficient. It necessitates the creation of a physical model (whether based on a plaster mould or digital scan) and there are multiple steps in the pipeline, each of which takes time. The aim of this project is therefore to make this process more cost-effective and efficient by eliminating steps 3 through 5. This is accomplished by developing a tool that allows a surgeon to determine the optimal fit entirely within a digital environment, obviating the need to produce a 3D model and thus eliminating redundant work.

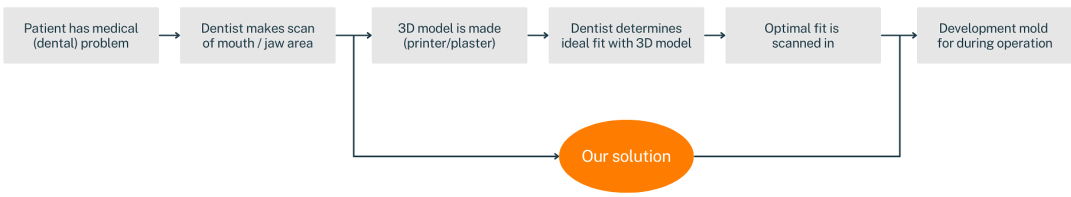


Figure 1: The current workflow.

During the problem description phase, Relu highlighted the following three key aspects that a potential solution must meet. Firstly, the product should exhibit as gentle a learning curve as possible, so that dentists and oral surgeons can easily switch to this new technology. Secondly, the user experience should be pleasant and feel natural, so that the specialist can find the optimal fit. For this reason, the jaws must appear as realistically as possible to the user, who should be able to see, move and potentially feel them. Lastly, specialists usually desire to have the final say on the details of a procedure. Thus, if any support is provided when making the fit, it must not hinder the specialist from making additional adjustments. All of these factors were considered when devising a solution.

3 Methodology — Thomas Speltincx

To support this project, multiple workshops on useful teamwork topics were given throughout the academic year. Every aspect and method taught was put to good use during the project. The given workshops are the following:

- A workshop on how to give feedback to each other in a constructive and helpful way.
- A brainstorming workshop where various techniques were introduced to help the team come up with ideas collaboratively and in a structured way.
- A workshop on coming up with a business model canvas in the form of a Lean canvas, helping the team deconstruct their problem and solution into a one-page plan.
- A pitching workshop, teaching the team ways to present and sell an idea to important stakeholders.
- An Agile Development workshop which introduced the organization and workflow of a software development project. This topic is further expanded on in section 3.1.

3.1 Agile development

Agile development [3] is a software development methodology that emphasises collaboration, flexibility, and rapid iteration. It is based on the idea that software development is an iterative and collaborative process that requires continuous feedback and adaptation. This approach emphasises the importance of individuals and interactions, working software, and responding to change. The goal of Agile Development is to deliver high-quality software that meets customer needs quickly and efficiently.

Scrum, a tool of Agile Development, involves breaking down complex software development projects into smaller, more manageable components called sprints. Each sprint is a short, time-boxed period of development, in this case two weeks long, during which a specific set of features or functionalities are developed, tested, and delivered. At the end of each sprint, a working software product is demonstrated between subteams (see 3.2), and feedback is gathered. Before every project session a stand-up meeting of maximum fifteen minutes is held. Each subteam prepares a short, abstract overview of their recent progress and difficulties, so the whole team is up to date and independent development can continue. The overall project, structure of the sprints and their deadlines were managed with Jira Software by Atlassian [4], a web app providing a project overview to all team members.

The overall use of Scrum worked out really well during this project, the method being very clear and natural to work with and pushing all team members to finish all goals and deliver a working product by the end of each sprint. The stand-up meetings were time-efficient and kept a healthy pressure on the entire team, once again pushing the project forward.

3.2 Team division

Teamwork is often key to a project's success, as the way a team is partitioned can make a significant difference in achieving set goals. In this software development project, subteams were used to enhance the overall productivity during sessions, by working in parallel.

Usually when a team is responsible for the development of a new software product from scratch, it is split in front-end and back-end development. This project, however, required two extra subteams: one for developing the VR environment and another for seamlessly integrating this VR environment in the web app. The nine team members, each with their unique set of already acquired skills, were thus divided into four dynamic subteams, as seen in Figure 2 in Section 4.

Team Front end Focused on developing the front end of the software, designing and implementing the web app and the user interface.

Team Back end Focused on developing the back end, the software's underlying infrastructure, including the database and the server-side logic.

Team VR Focused on developing the VR environment, implementing user functionalities and modeling dental occlusion as realistically as possible.

Team Push Focused on finding a way to develop the ability to transition between web app and VR, including data transfer and user convenience.

Of course the teams were not fixed during the entire project. The second semester was started with a proportional distribution across subteams, but it was clear after the first two sprints team members that had to be shifted, as the Push Team had done its job and help was needed in the VR Team. Assigning extra people to the more difficult VR development worked out great as the tasks were relatively easy to divide and the project thus progressed more quickly.

As an overall reflection, the way the team was divided worked out great for several reasons. First, it allowed each team to focus on their respective areas knowing the other teams were doing the same, which enabled the team members to work more efficiently and effectively, without getting distracted. Meanwhile, everyone was kept up to date thanks to the weekly stand-up meetings, as described in Subsection 3.1.
Secondly, the team division encouraged collaboration and communication. By breaking the team into four subteams, each with a clear focus, the team members were able to work more closely together and share ideas and feedback in a way that enhanced the quality of the software.
Thirdly, the team division allowed for rapid iteration and progress. By dividing the project into manageable chunks using Scrum, the team was able to make progress quickly, with each sprint delivering new features and functionalities that could be tested and refined.

4 Tech stack

4.1 Overview — Anna Proost

A first version of the Tech Stack was established at the end of the first semester. Throughout the actual development of the project in the second semester, changes were made as issues with certain frameworks came to light and new possibilities of other frameworks were discovered. For example, it was decided within the front-end team that Bootstrap had a limited functionality and that it would be easier and more attuned to Relu's UI to write the CSS by hand. Thus, Bootstrap was taken out of the Tech Stack.

Figure 2 shows the final version of the Tech Stack. The following sections will discuss the used technologies in further detail, reasoning why every framework was chosen as the best option to employ during the project.

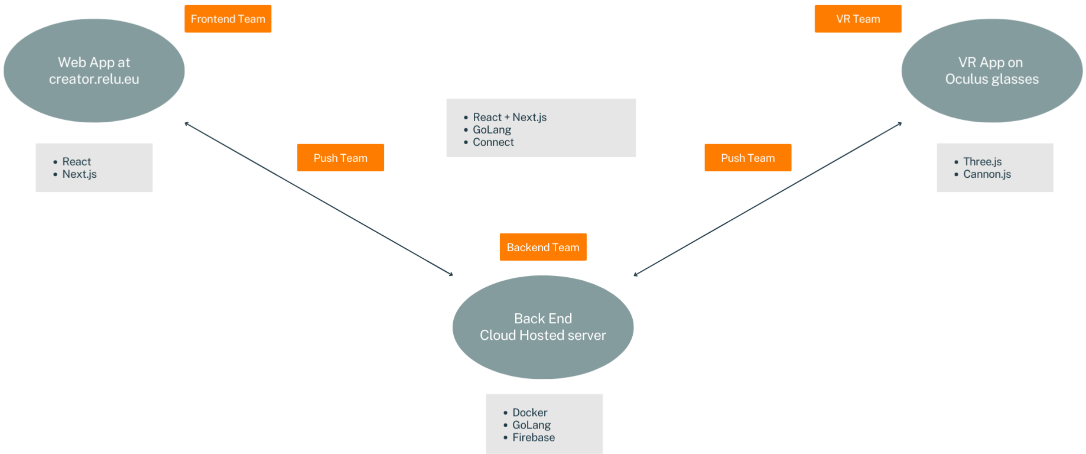


Figure 2: The project's final Tech Stack

4.2 GoLang — Lars Depuydt

The back end of the application is written in GoLang, also called Go. As stated by Yaroslav Bai [5], Go is meant to simplify the process of software product development, particularly for complex architectures and processes.

The Go language has three main advantages. First of all, Go is fast. This is because it's a compiled language, meaning code is directly compiled to processor code, without needing an extra layer like Java bytecode. Second, Go is a simple language with the same procedural approach as C and Java, making it very easy to learn. A third key advantage of Go is its ability to scale and support concurrency through Goroutines. These are essentially functions that can run simultaneously and independently. They are non-blocking by nature and take up only a very limited amount of memory, which makes them perfect for scaling.

However, Go also has some drawbacks. One such thing is that writing code in Go can be a time-consuming process. Due to the strict and minimalist syntax, it may require more lines of code than for example a language like Python to achieve the same functionality. Second, Go doesn't have support for generic functions. Generic functions are a collection of different functions with the same name, but with undefined types of inputs during compile time. Types are then assigned during compile time. Not having this functionality can reduce code efficiency and reusability.

The decision to use Go for the back end was based primarily on its simplicity and ease of learning. Given the limited time span for the project and varying levels of coding experience among team members, it was important to use a language that had a short learning curve. Go's straightforward syntax and minimalist approach made it a suitable choice for a small demo project.
Additionally, Go's focus on clean and readable code, enforced through strict formatting rules and lack of

unnecessary features, made for another advantage. This feature is particularly important for projects with multiple contributors, where maintaining code consistency is essential.
Lastly, considering Go's growing popularity and adoption in the software development industry, learning it is certainly not a wasted effort. According to Github Language Stats [6], Go is one of the fastest-growing programming languages, with a steady upward trend in recent years. This suggests that learning Go could be a valuable skill for future projects or job opportunities.

4.3 Docker — Lars Depuydt

To make the development process easier and deploy the app, the project makes use of Docker. As stated by Mounika Narang [7], Docker is made to simplify managing the application dependencies and technology stack across different devices and environments. This is a common difficulty in modern app deployment. To do so, Docker uses Docker containers to simplify the building and operation of an application. These Docker containers are standardised executable components, which link application code, operating system libraries, and all the dependencies needed.

An additional advantage of Docker is isolation, as Docker containers are completely self-contained. This improves the reliability even more, because it can be guaranteed that code in production will run in exactly the same environment as during testing.
Docker containers are different from virtual machines because they are an abstraction at the operating system level. This leads to a modularised functionality so that components can be deployed, tested, and scaled on their own. They are also more lightweight and require less startup time than virtual machines.

Docker containers can be described with a Docker file, which contains the commands required for creating a Docker image. A Docker image contains everything needed to run the program. With it, it can be successfully deployed everywhere on a Docker instance. To enable communication between Docker containers, Docker Compose is used. This is a service that allows several containers to run simultaneously while a network between them is established. For example, the front-end Docker container needs to communicate to the back-end container through the API, and this container needs to communicate with the database container to be able to get the persistent application data. These connections require a network connection. All this is specified in the Docker compose file of the project, which is written in YAML.

4.4 Google Cloud Platform & Firebase — Dag Malstaf

The project leverages the capabilities of Google Cloud Platform (GCP) and Firebase to efficiently deploy and manage the back end and front end of the developed application. GCP is a suite of cloud computing services that provides a variety of tools for developers to build, test, and deploy applications. Firebase is a platform developed by Google that offers a range of services to facilitate web and mobile app development, including hosting, real-time database management, and authentication.

The choice of GCP and Firebase as integral parts of the tech stack is based primarily on their ease of use, scalability, and seamless integration with other Google services. In the project, GCP's Cloud Build is used to automatically rebuild the back-end container whenever a new update is pushed to the main branch on GitHub. This feature ensures that the application remains up-to-date and minimises manual intervention, streamlining the development process. Cloud SQL, another GCP service, is utilised for managing the user data database. The Cloud SQL proxy is employed to link the back-end container to the database, providing a secure and efficient means of managing data.

Firebase played a crucial role in deploying the front-end application. For rapid deployment of the Next.js app, Firebase Hosting was used. This in combination with Firebase Functions, enabled the development of serverless functionality. Firebase Storage was utilised for storing large 3D files, which are central to the application. These services offered by Firebase not only simplified the development process, but also ensure a high level of performance and security.

The combination of GCP and Firebase as part of the tech stack aligns well with the project's requirements. The platforms provide a solid foundation for the application, allowing us to develop a digital tool that streamlines the oral surgery treatment process, reduces costs, and improves overall efficiency. The simplicity, scalability, and seamless integration of GCP and Firebase services, combined with their compatibility with other components of the tech stack, make them an ideal choice for meeting the needs of the project and its end-users.

4.5 Next.js — Arne Van Damme, Anna Proost, Astrid Dewever

The user interface of the project is built using React, a library for JavaScript. React allows developers to write HTML-like syntax in JavaScript, which is then combined with CSS from a separate file to render the user interface. It uses hooks to communicate with the back end and supports events and event handlers.

Besides React there is an immense amount of other possible languages that can be used instead, the common denominator being that they all use a modular architecture, where the focus lies on user-built *components* that can be reused in multiple places of the interface. Almost all of these front-end frameworks are also considered reactive, which means the user interface is rendered whenever the state of a component implemented in it changes.

Examples of other front-end frameworks are Angular, Vue and Svelte. Angular is mainly focused on building large complex web apps, which also entails that it has a very steep learning curve. That did not seem appropriate for the current project. Vue on the other hand is more light-weight and combines CSS, HTML and JavaScript in the same file. It is connected to the back end via a REST-API, like most other frameworks. Svelte was made to reduce the amount of redundant code that had to be written by developers. It is generally considered a very compact and user-friendly framework. Unlike the other frameworks mentioned above, Svelte compiles its code to imperative JavaScript language that works in the browser.

Despite the existence of the frameworks available, React was chosen for the project due to it's popularity, which results in the existence of many tutorial videos, templates and general help with common errors. It is also heavily demanded by employers and thus a valuable skill to develop. Later in the project, the choice for React proved to be a sound one for yet another reason, namely the other libraries' compatibility with it. E.g. all input forms in the web application are made and managed with the form library Formik, which was especially developed as a React library.

Following this decision, Next.js was chosen as the React Framework to use. Some of the reasons were its server-side rendering (SSR), which makes for fast load times and a great user experience, as well as the large amount of existing compatible modules and plugins. Moreover, Next.js has a very easy and intuitive routing system based on pages. Files added to the page directory automatically become available to the (static) routes. This way of routing extends to dynamic routes when needed, filling in *slugs* or parameters at request time. This way of routing turned out to be ideal for linking to different dentist's and patient's dashboards.

4.6 Three.js — Kjell Mesotten

The VR portion of this project is implemented using the web-based 3D JavaScript library Three.js. Before landing on this technology, many other options were considered.

The first decision that had to be made was choosing between a web-based virtual reality development framework or a game engine. Game engines tend to have a lot of positive aspects, such as very high-quality visuals, plenty of documentation and intuitive user interfaces. However, their negative aspects were too significant to ignore:

- Absence of web support: this would require a large, dedicated program installed on the VR-headset, which was deemed undesirable for the project.
- Limited collaboration tools: many game engines, like Unity and Unreal Engine, have limited support for cooperation and version control.
- Performance: game engines require a lot of overhead, which reduces performance. This is especially true for a more lightweight application like the one in this project.

For these reasons, a web-based 3D Virtual Reality framework was preferred. Three of such frameworks particularly stood out. What follows is a short summary of the pros and cons of each.

PlayCanvas

+ Excellent collaboration tools
+ Instant testing on VR headset
+ Easily integrates 2D and 3D user interfaces

- Pay-per-seat, otherwise limited to public projects
- No external hosting possible on the free plan
- Very limited and incomplete documentation

Babylon.js

+ Open-source
+ Typescript instead of Javascript
+ Solid built-in VR support
+ Excellent debugging tools

- Fewer tutorials than Three.JS
- Produces CPU-intensive software
- Difficult to set up

Three.js

+ Open-source
+ Extensive documentation
+ Produces more lightweight applications
+ Fast and efficient

- More geared towards 3D visuals than VR interactions
- Lots of additional libraries required
- Difficult to test on VR-headset

A fourth framework, named A-Frame, deserves a mention. It is built on top of Three.js, but its main focus is Virtual Reality, which seemed perfect for this project. However, it soon became apparent that A-Frame is very restricting and would not allow proper collision detection of the jaws to be integrated.

PlayCanvas was quickly found to be a bad choice for the project, as it was impossible to create a private project without paying additional fees and the documentation was very limited. Babylon.js was also removed from consideration, because it was difficult to set up. This would have cost us valuable time that was better spent on development. The VR portion of the project was thus created using Three.js.

4.7 Cannon.js — Senne Vanden Eynde

While Three.js is able to handle the visualisation aspect of the VR environment, the project also requires a way to account for the physical interaction between the jaws. For this, various physics engines were considered as they have built-in collision detection. The engine used for the project would need to be lightweight, since the application would essentially be running in the browser of the VR headset, and the computational resources would thus be limited. With this requirement in mind, various physics engines were considered, given their inherent capacity for collision detection. Of particular importance was the selection of an engine that is able to operate efficiently within the constrained computational resources of the VR headset's browser environment. As such, the chosen engine would need to exhibit a lightweight profile to ensure optimal performance.

There are not many 3D physics engines for the web, however, so only two options were considered: Ammo.js and Cannon.js. Cannon.js was ultimately chosen for the following reasons. First, Cannon is more lightweight and easier to implement [8]. Second, it is written directly in JavaScript instead of being ported from a C++ library like Ammo.js, so it was assumed Cannon would be easier to extend and integrate into the web application. It turns out Cannon also has an actively maintained fork *cannon-es*, which is written in Typescript. This integrates well with Next.js, which was used in the rest of the web application. Third, Cannon works well in combination with Three.js since it allows making physical objects, so called Cannon bodies, out of the meshes Three.js provides. These can then be used in the physics simulation to detect collision.

5 Functionality

5.1 Overview — Anna Proost

As previously stated, for the development of the project both the task and the team were separated into four big units: front end, back end, push and VR. In the following sections, each unit's activities and outcomes will be discussed in depth, while still maintaining a reasonable level of abstraction so as not to get lost in technical details.

Before starting the actual implementation, towards the end of the first and the beginning of the second semester, a number of flowcharts and visuals were created. These diagrams, depicted in Figures 3 and 4, were intended to provide a general overview of the project to all team member and to get a deeper understanding of its structure and the different functionalities that were to be included. As the project progressed, those flowcharts were continually updated to reflect the latest set of features. This practice allowed the whole team to keep abreast of the planned implementations.

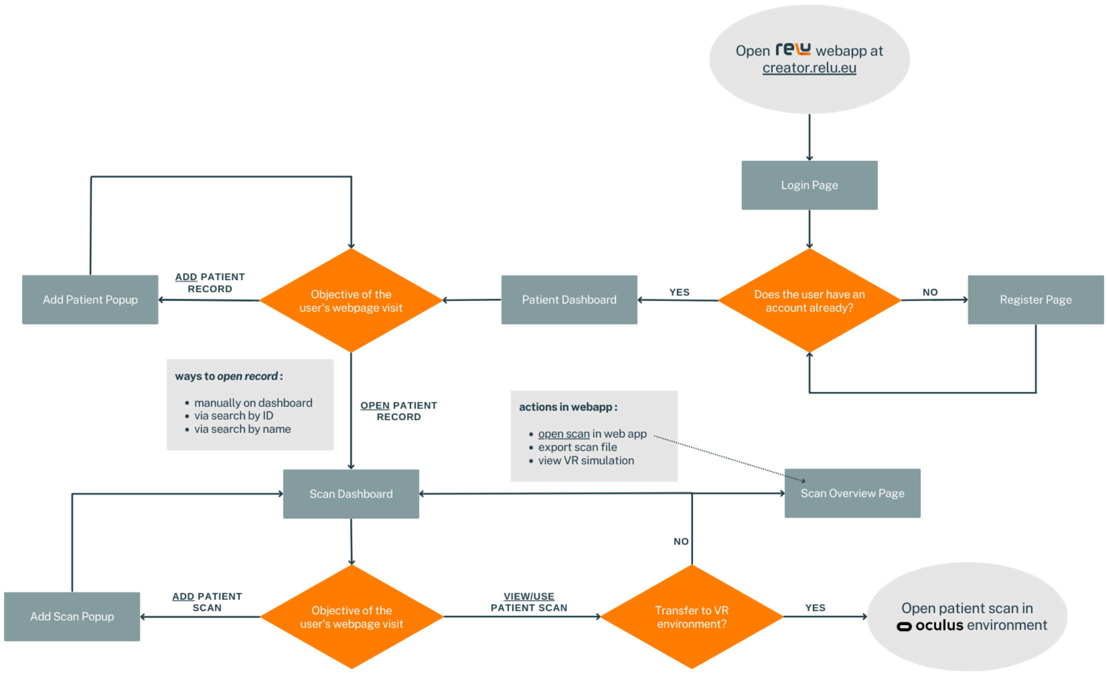


Figure 3: The web app flowchart

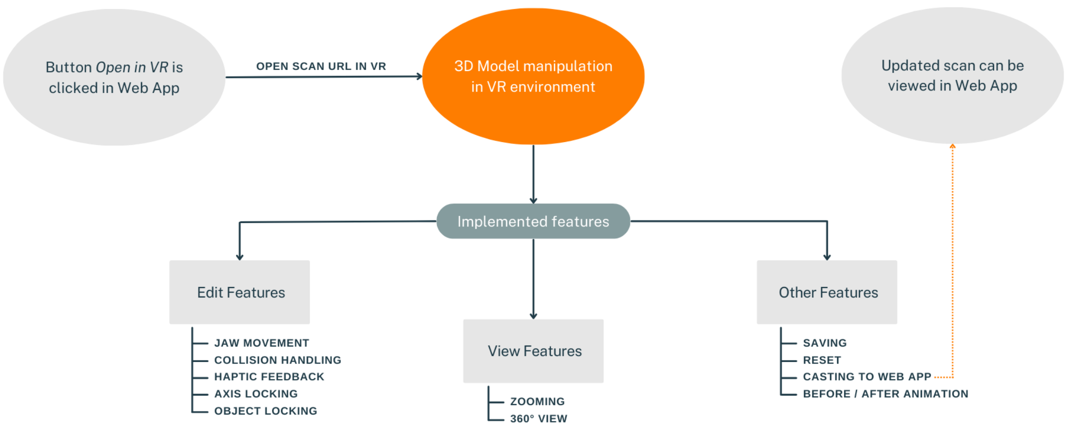


Figure 4: implemented VR features

5.2 Back end

5.2.1 Database — Lars Depuydt

The back end employs a Postgres database to ensure persistence of all data. Postgres is a type of SQL database, which implies that the data is stored in a relational schema. This schema is depicted in Figure 5.
The schema defines four different entities, namely 'dentist', 'patient', 'scan', and 'scan_save'. The dentist entity represents the user of the platform, and their login credentials are stored within the database. Next, every patient entity is linked to a dentist through their dentist's id. These ids are automatically incremented. Subsequently, each scan is associated with a patient. The scan field only stores the URI of the actual scan file, while the file itself is stored within a bucket on the cloud. The created_at field of scan is automatically added when an entry is created. Finally, each scan has one or more scan_save entries, which contain a timestamp and a set of coordinates for both jaws. This schema fully captures all the necessary data for the smooth operation of the entire application.

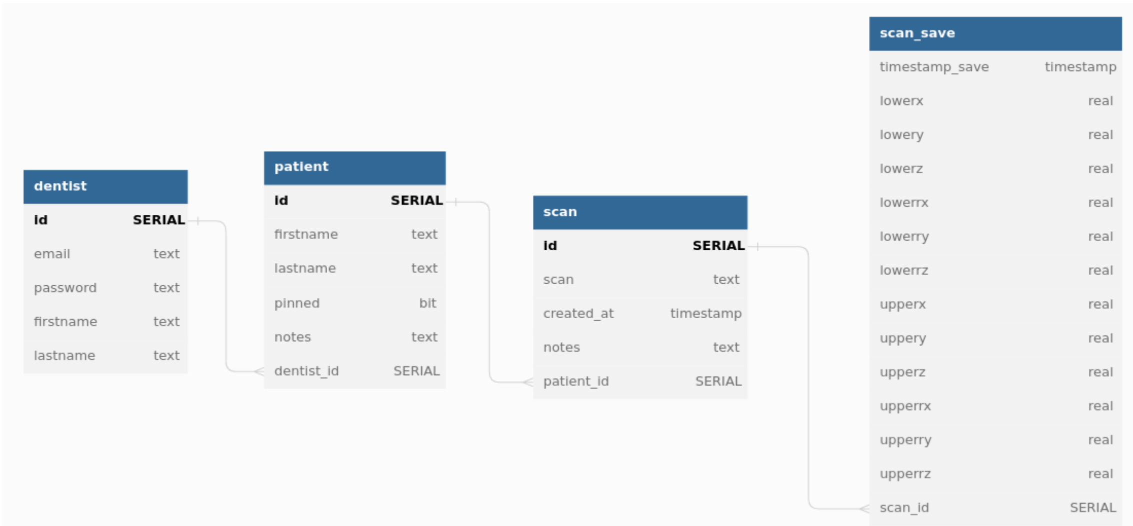


Figure 5: Datebase schema, made with DB Diagram [1]

5.2.2 The API with Connect — Lars Depuydt

To communicate between the front end and the database, a back end is created. This back end provides an Application Programming Interface (API), which is used to interact with the back end. The back end then communicates with the database, where data is stored persistently. This API is written using Connect, a part of the Node.js framework.

Connect is a collection of libraries for building browser and gRPC-compatible APIs [9]. The structure of the API is written using protocol buffers. These protocol buffers are a language-neutral, platform-neutral, extensible mechanism for serializing structured data [10]. In this file, it is defined how the data needs to be structured. This results in a typed API, because the proto files can be used to generate the appropriate API files for each language. Connect handles this automated API implementation, so that interacting with it from Typescript in the front end or from Go in the back end is very clean, straightforward, and reliable.

The main advantage of using protocol buffers over a RESTful API, is that the API's input and output types are guaranteed. As stated in the documentation of Buf [11], in a REST API, there are no rules, no standards, and no enforcement of any kind. It just comes down to hoping that users won't break these rules by reading the documentation carefully. Building a protocol buffer is rather hard and time-consuming. This problem is solved by Buf and nicely implemented in Connect.

The dependencies needed for the generation of these files were added inside a Docker container, refer to section 4.3. When set up correctly, this allows for a very stable, reliable, and fast implementation of an API that can easily be used right in every context.

5.2.3 Communication Web app - Database — Anna Proost

As mentioned above, the communication between front end (the web application) and back end is done via an API, allowing for abstraction on the front-end side - just simple function calls that return the information needed or deliver a success message in case of a changed database table - while all technicalities and database manipulation is handled in the back end. A full listing of all calls - along with their needed attributes and the elements that are to be returned, can be found in the `service.proto` file, which will update itself from time to time when changes were made on the back-end side.

The API calls can be done either automatically, e.g. when a new page is loaded, or on a certain action, like the click of a button. The biggest challenge in the connection of front and back end turned out to be exactly this fine-tuning of the API calls that made changes to any database tables, so that only fully filled-in and intended calls could be made.

An elegant solution to this complexity was achieved by implementing extra React state variables, `[sumbitOK, setSubmitOK]` and `[sendOK, setSendOK]`, in the front-end code. These state values would indicate when all info was completed and submitted, and when API calls could be made safely, respectively. Along with this, a few other features of the React library were used. One of these was the `useEffect()` hook [12], which is automatically called when changes are made to any of its given dependencies (e.g. the back-end-generated response `data` or any of the aforementioned state values), executing a defined callback function in case of those changes or general re-rendering of the page.

Another feature used were the functions included in the react-query library, which allowed the front end to define a query connected to an API call, and use it separately and only when `refetch()` hooks were called elsewhere in the code. All together, those functionalities allowed for the front end to be connected correctly to the database, sending the correct information at the right time instead of overwhelming the back end with unfinished or useless calls.

5.2.4 Communication VR glasses - Webapp — Kjell Mesotten, Senne Vanden Eynde, Jonas De Schouwer

Upon first use, the desktop and the VR goggles need to be paired using an automatically generated code. The code is shown on the desktop and needs to be entered on the headset, as shown in Figures 6 and 7. Submitting this code will pair the devices together for all subsequent sessions. Through the use of cookies, this process only needs to be done once.

An automatically generated code has been chosen as authentication mechanism rather than a login on the glasses for the customer's ease of use. After all, it is quite difficult to type on such devices. It is assumed that this approach is sufficiently secure for an MVP.

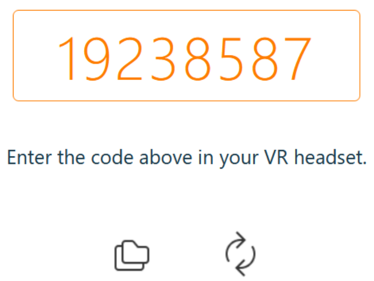


Figure 6: Settings page on desktop showing the generated code

Once paired, the application on the VR headset starts waiting for messages from the desktop. As soon as the user presses the button on their computer that opens a scan in VR, the URL of the correct scan is sent to the

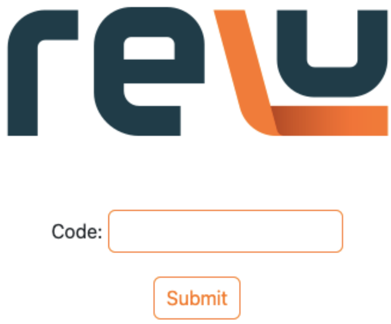


Figure 7: Webpage on VR goggles

headset. The application is then redirected to this URL and the user is immersed in an interactive environment, in which he or she can examine and manipulate the scan. One can also choose to open the VR application after sending the scan from the desktop page. If any messages to redirect were received, it will immediately do so. After the user has finished editing the scan, they can exit and the final position of the jaws will be stored in the database (more details in 6.2). The desktop will be notified when the VR device exits their session.

Like the rest of the back end, communication between the VR goggles and desktop is implemented in Golang with protocol buffers. Go channels allow the paired desktop and VR devices of all users to communicate messages to each other on the server.

5.3 Front end — Arne Van Damme

As was already mentioned in section 4 the front-end part of the MVP was built with React, CSS and HTML. Extensions like Formik and Next were used as well.

The user interface consists of four main pages. The first two are the login page and the register page, where users must authenticate with their e-mail and password. Logging out can be done at any time with the log-out button in the header. The other two pages are the patient overview and the scans dashboard. In the patient overview, dentists can search through a list of all patients by entering their name or ID. Patients can be added to and removed from this list. When added, all the information of a patient must be provided, if needed, a patient can also be edited afterwards. Pinned patients will be displayed first in the list, at the top of the page. Notes can be added to patients with the edit button.

Each patient has their own separate list of scans, these can be seen in the scans dashboard. Just like patients in the patient page, scans can be added, removed and edited. Users can also assign tags to scans to get a quick overview of all possible issues. Examples of these tags could be: crossbite, overbite, underbite, etc. These tags can then be used to filter through the scans with a button in the sidebar. Every scan is equipped with a button to enlarge the picture on the computer, so the dentist and patient can both look at the scan to discuss possible treatments. The VR button is the most important button on each scan. It takes the user to the VR environment where they can modify the occlusion. This is discussed further in section 5.4. Another provided feature is the possibility to make an animation of the occlusion for the patient. This shows the progression from the initial scan to a fitting occlusion. Scans can also be downloaded from the application, so dentists can share them via e-mail.
Aside from the four main pages that were just discussed, there is also a settings page. This page contains the code that connects the VR headset to the computer. The linking process was already mentioned in section 5.2.4.

Since there are so many buttons, a patient module, just like a scan, has three different windows layered on top of each other. The first one simply displays the name of the patient and a picture with some additional side info. Hovering over the patient module will reveal the second window with the most important buttons, like open, delete and pin for example. There is also a button in the top right corner that opens up the third and last window. This is a drop-down menu with the less important buttons.

To keep things neat, the patient page and the scans dashboard are both provided with a sidebar and a header, containing the log-out button. The sidebar displays a greeting to the user, and has a list with commonly used tools. At the bottom of the sidebar there are some navigation buttons to other pages, including a link to the official Relu website.
Some buttons, like the edit button for example, open up a pop-up window. This is called a modal, it renders all other tools of the user interface unclickable until the pop-up is closed. Like the different windows discussed above, this is used to allow more actions within the same page, even though the screen area is limited.
For the best user experience the amount of clicks needed to do the most important operations are limited. That is also why the less important buttons are more hidden away behind drop-downs and pop-ups. This leaves space for more important actions to be immediately clicked or used.

The front end is connected to the back end by the means of routes and hook calls. This was further explained in the back-end section 5.2. Between the different pages of the user interface, some information is also remembered. This allows for greeting the dentist for example. It is needed for remembering which scan of which patient was opened as well.

5.4 VR

5.4.1 Approach and specifications — Kjell Mesotten

The implementation uses the *Meta Quest 2* VR goggles (also referred to as *Oculus Quest 2*) with two controllers, and it is hosted on a web page. The scan of a patient's jaws can then be rendered in a Virtual Reality environment, and the controllers can be used to interact with it. The lower and upper jaw are separate objects and can each be moved independently with one of the two controllers, while the software will ensure that both parts of the teeth cannot overlap in the three-dimensional rendered VR space. If the upper jaw touches the lower jaw in the simulation, the controllers will vibrate to indicate to the user that further movement is not possible.

An important design choice that is made is the type of application that runs on the VR headset. Three choices were examined, including a native application on the VR headset, a Progressive Web App (PWA) and a web page. A PWA was ultimately chosen on the following grounds:

- For reasons outlined in section 4.6, the VR application is hosted as a web page by default. This would mean a native application waiting for a request should open a browser when the user wants to open a scan in VR. This takes longer than a simple redirect in a browser or PWA.
- A PWA is listed as a VR app in the application list. This is easier to open than having to enter a link in a browser (as with the web page).
- A PWA is easier to implement than a native Meta Quest 2 app.

The PWA was created using *Bubblewrap CLI.* When it is opened by the user, the application navigates to the web page where a waiting page is hosted. When the user chooses to open a scan, this page will be redirected to the VR environment.

Based on information from Relu and various interactions with potential users, many features were selected that may be useful in the VR environment. The following list provides an overview of the features that have been implemented. Section 11 will take a closer look at potential future features that were deemed out of scope for this project.

360-degree view The user can walk around the jaws to get a full picture of the situation.

Jaw movement The jaws can be grabbed and moved around with the VR controllers.

Collision handling Just like with a physical mould, it is not possible to move the jaws through one another.

Haptic feedback On contact, the controllers vibrate, letting the user know that the jaws cannot move any further.

Axis locking When the X button is pressed, the movement of the jaws will be limited to the direction of a certain axis.

Object locking When a jaw is not selected, it is locked in place.

Zooming The user can make the jaws larger or smaller, so that the jaws can be positioned with sufficient precision in relation to one another.

Undo/Redo The user can revert all movements by pressing the undo button and can return to the most recent movement by pressing the redo button.

Saving The relative jaw position is saved automatically every sixty seconds and uploaded to the database. It is also possible to save the jaws manually, by selecting the save option in the menu.

Before/after animation A short animation showing the change in jaw position that can be of interest to both the surgeon and the patient.

Reset When the reset option is selected in the menu, the jaws move back to their original position.

Casting Display the VR view on the desktop.

The following subsections explain the different data structures and take a closer look at the two core features: collision handling and jaw movement.

5.4.2 Data structures — Jonas De Schouwer

As mentioned in subsections 4.6 and 4.7, Three.js and Cannon.js were used to implement the VR environment. Three.js is responsible for visualizing the jaws and interpreting controller and headset inputs and Cannon.js takes care of all physical interactions in the scene.

The main data structures used for the VR environment are summarised in Figure 8. First of all, the inputs are the headset and controller positioning and the controller buttons that are pressed. All of these are registered and handled by Three.js. Second, the relevant Three.js data structures are the jaw meshes, the controller meshes and the targets. The jaw meshes are the 3D triangle models derived from an intraoral scan, they are what is displayed to the user. The controller meshes are also displayed on the appropriate location for a convenient user experience. The last three.js data structure is a 'target' per jaw that contains the desired jaw positioning, based on the controller input. Finally, the only relevant Cannon.js data structures are the two jaw bodies. These are simplified versions of the jaw meshes, which are used for collision detection. When selected, the bodies are moved to their respective targets with impulses and torques.

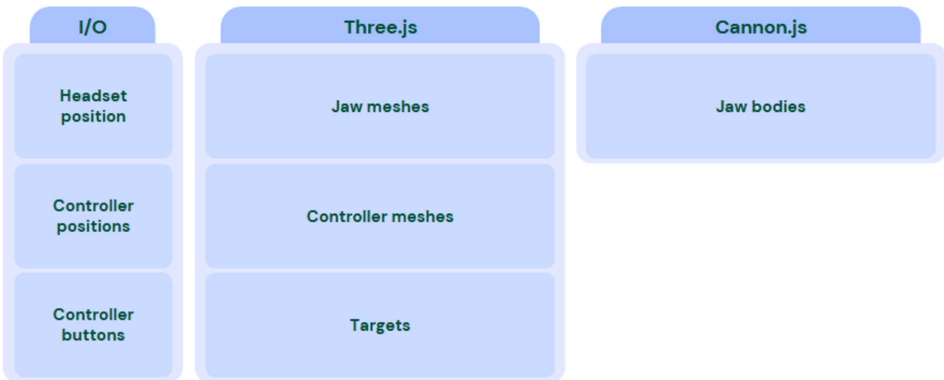


Figure 8: VR data structures

5.4.3 Collision handling — Jonas De Schouwer

As mentioned above, Cannon.js is responsible for the physical interactions in the simulation. Accordingly, the collisions between both jaws are handled by its built-in collision detector and solver, with some adjustments.

Both of these are enabled by default in Cannon.js as long as the bodies involved are convex. The jaws in the simulation, however, need a distinctly non-convex shape to find a fitting occlusion. Hence, the jaws are first decimated to obtain a model with a lower triangle count and then decomposed into a set of convex meshes using the HACD algorithm [13]. This decomposition is precomputed and saved in the backend. The different steps in this approximation are visualised in Figure 9. Subfigure 9a shows the original scan, 9b displays the model after decimation and 9c visualises the convex decomposition.

(a) original intraoral scan (b) decimated scan (c) convex decomposition

Figure 9: Approximation of the intraoral scan in two steps

The approach described so far works, but lacks performance. This is because the Cannon.js collision detector executes the computationally intensive function *findSeparationAxis* for every face of both jaw models, which leads to framerates around 2 fps on the *Meta Quest 2* and thus an unnatural sensation. Hence, it is necessary to use Javascript Web Workers for pulling this work out of the main thread. This is enough to increase the simulation's performance to a visually pleasing level.

5.4.4 Jaw movement — Jonas De Schouwer

In order to move the jaws, it is not allowed to simply set their position as this might induce an overlap and thus a glitch. Instead, the jaw movement is controlled by impulses and torques in the right direction. Concretely, a 'target' is created for each jaw, which holds the desired position and orientation. An impulse and torque proportional to the positional and rotational difference between the jaw and the target is then applied on each frame. Furthermore, the jaw velocity and angular velocity are heavily damped to reduce overshoot.

6 Results

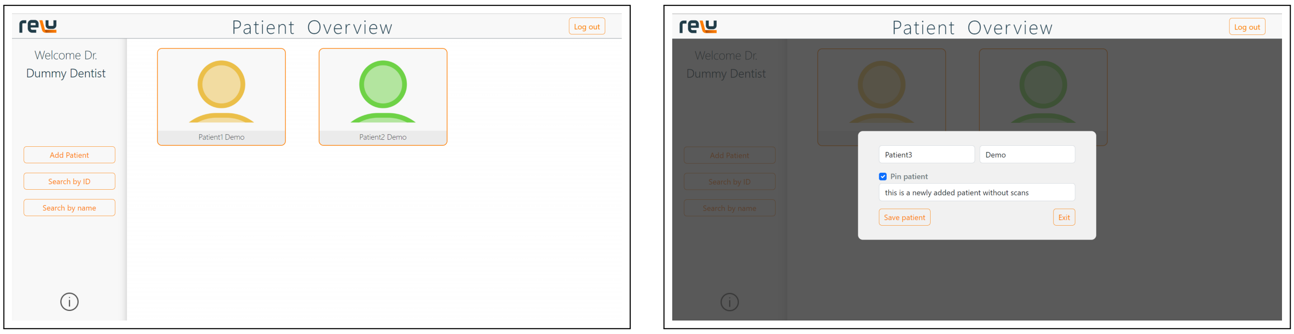
6.1 Web app — Anna Proost

In this section a run-through of the finished web app and its implemented features will be shown, based on the flowchart in figure 3. Along with that, further down in this section, there will be a short discussion of the non-functional choices made during the development and the thought process behind them.

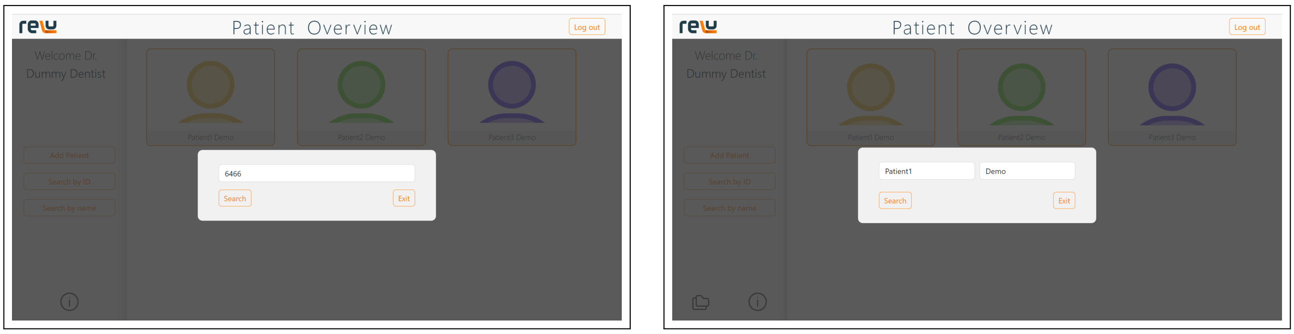
(a) Login Page (b) Register Page

Figure 10: User opens the web page

The run-through starts with the user opening the web app at creator.relu.eu, which takes them directly to the Login Page (10a). There, the flow diverges, depending on whether the dentist already has an account with Relu. If so, they can simply put in their credentials and log in. If there is no record of their account in the database, they can click to the Register Page (10b) and request one by filling out their information. Clicking the *Register* button takes them back to the Login Page once again, where they can now log in.



(a) Patient Dashboard (b) Add new patient



(c) Search by ID (d) Search by name

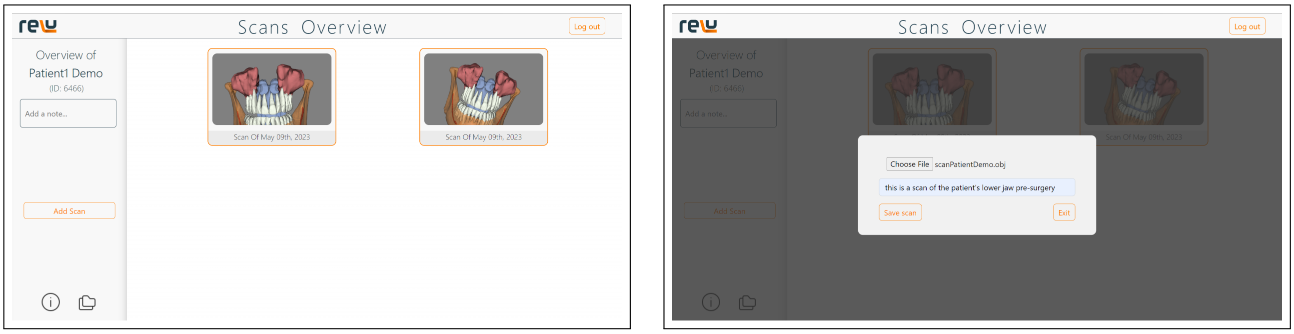
Figure 11: Options on Patient Dashboard

Once successfully logged in, the user finds themselves on the Patient Dashboard (11a), where all of their patients are listed. Like mentioned in the flowchart, there are two major options.

First of all, the dentist can add a new patient to their dashboard by clicking on the *Add patient* button in the sidebar, which opens up the popup seen in figure (11b) where they fill in the requested information. By clicking the *Save patient* button, the popup window closes and the patient will appear on the Patient Dashboard immediately (and as the very first if it was pinned). If the dentist presses the *Exit* button instead, the action is terminated, meaning the window closes without saving the patient info or adding it to the database.

The second action is opening up a patient's scans. This can be done in multiple ways. The first one is to simply scroll through the dashboard until the desired patient is found and opening their scans by hovering over the picture and clicking the *Open* icon (as explained in section 5.3). If a dentist has a lot of patients however, they might want to find them in a handier way that doesn't require looking though the whole dashboard. For this reason, the methods *Search by ID* and *Search by Name* were developed, both called by pressing their respective buttons in the sidebar, which open up a popup window, requiring the following actions:

- In the Search by ID popup (11c), the dentist fills in the patient's assigned ID. Clicking on the *Search* button either opens up the patient's Scans Dashboard (see figure (12a)) or brings the user back to the Patient Dashboard if the patient doesn't exist.
- In the Search by Name popup (11d) both the first and last name should be filled in. Clicking on *Search* will open up a new Patient Dashboard with all of the patients with that name if there are any, or brings the dentist back to the general Patient Dashboard if there aren't.



(a) Scan Dashboard

(b) Add new scan

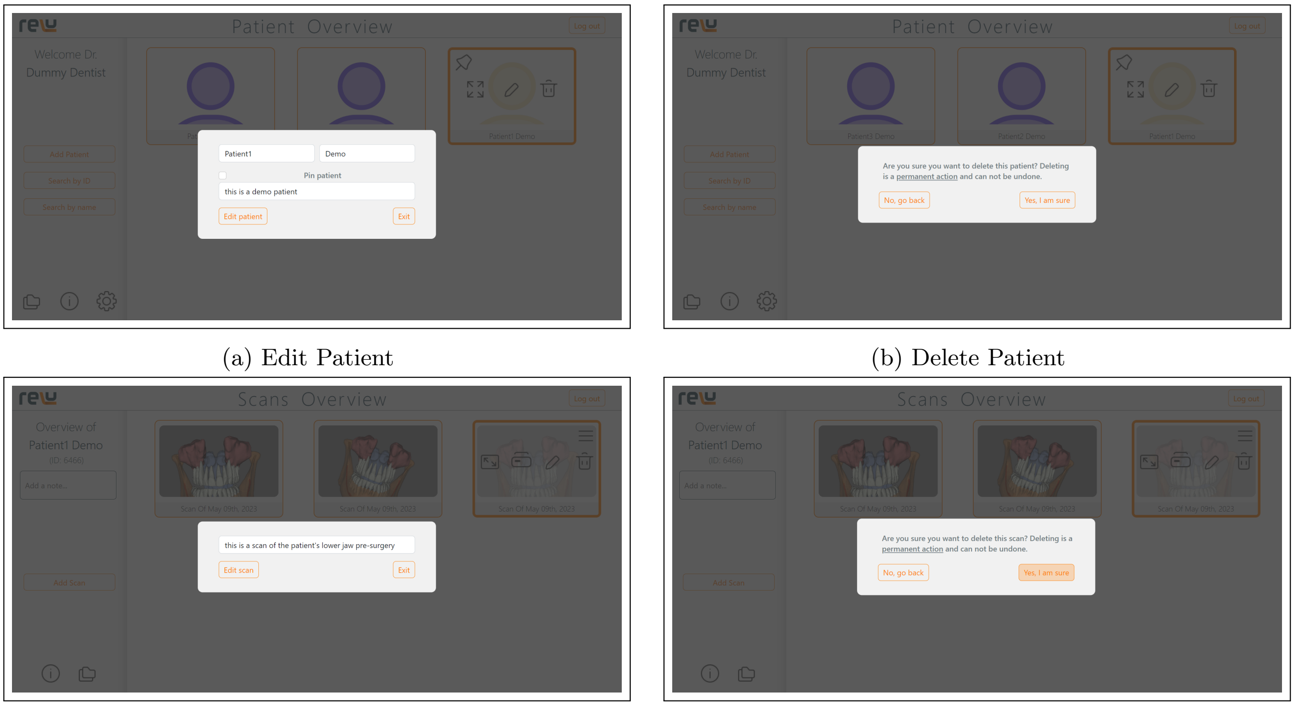
Figure 12: Options on Scan Dashboard

Opening a patient, no matter in what way, brings the dentist to their Scan Dashboard, with all scans that have been assigned to the patient over time. Here, a new scan can be added in a very similar fashion to how a new patient is added, as seen in figure (12b). Once added to the patient, the scan once again appears immediately on the dashboard.

Like explained previously in section 5.3, hovering over a scan will show a number of buttons to execute certain actions, which can result in the user being directed to different pages.

Figure 13: View scan in web app

Clicking on the *Open* icon will open up a web app view of the scan (13), which can be used if the dentist wants to quickly check the scan, show the patient their scan or a number of other reasons. On the contrary, if the dentist wants to work on the scan to find the optimal fit, this is done in VR and thus they should click the *VR* icon. When clicked, the scan will open on the connector Oculus glasses and the dentist can start manipulating the model in the VR environment. A detailed run-through of this is described in section 6.2.



(a) Edit Patient (b) Delete Patient

(c) Edit Scan (d) Delete Scan

Figure 14: Other patient & scan features

Although a number of other buttons and features were implemented, all of these have already been talked about in section 5.3, and thus describing their functions again in detail would be entirely redundant. As is clear from figure (14), both patients and scans can be edited and deleted, .

As previously mentioned, Relu had emphasised the importance of a gentle learning curve and an intuitive working environment, and thus a lot of importance was given to developing a simple no-nonsense web app. The main focus of the web app is user-friendliness and the easiness to get around it and through to the VR-environment, aiming to make the switch from physical to virtual fitting as smooth as possible for the dentist. For this reason, a lot of attention went to making sure the UX was comfortable and logical, with important actions clearly visible instead of hidden away.

A conscious effort was also made was to stay true to Relu's signature colours and design as much as possible, in an effort to promote uniformity with Relu's already available tools. A style guide (see figure (15)) was developed to keep the web design consistent, even with multiple members working on it at the same time.

Colour	RGB	HEX	Example	Use
blue / black	(32,60,72)	#203C48		Text colour
orange pop	(254,125,0)	#FE7D00		Signature colour
light grey	(230,230,230)	#E6E6E6		Popup colour
dark grey	(133, 156, 158)	#859C9E		Compl. to light grey

Figure 15: The colour style guide

6.2 VR environment — Senne Vanden Eynde, Thomas Speltincx

The VR environment is created in the application on the headset when the client sends the scan from the computer. The corresponding jaws then get displayed on a canvas. The user can enter the VR environment by selecting the *Enter VR* button with one of the controllers. This will display the canvas in full screen, enveloping the user.

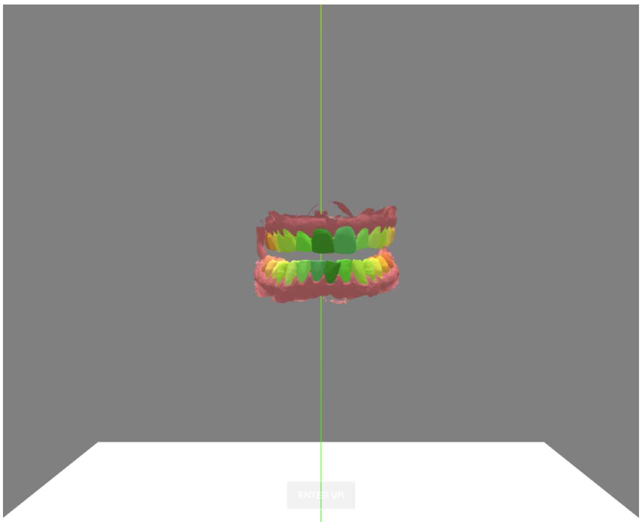


Figure 16: Canvas with jaws

The user can walk around to navigate in this environment and change their perspective by moving their head. Any movement of the controllers will also be visible in this space. The jaws of the scan can be interacted with in this environment using the controllers and their buttons, see Figure 17.

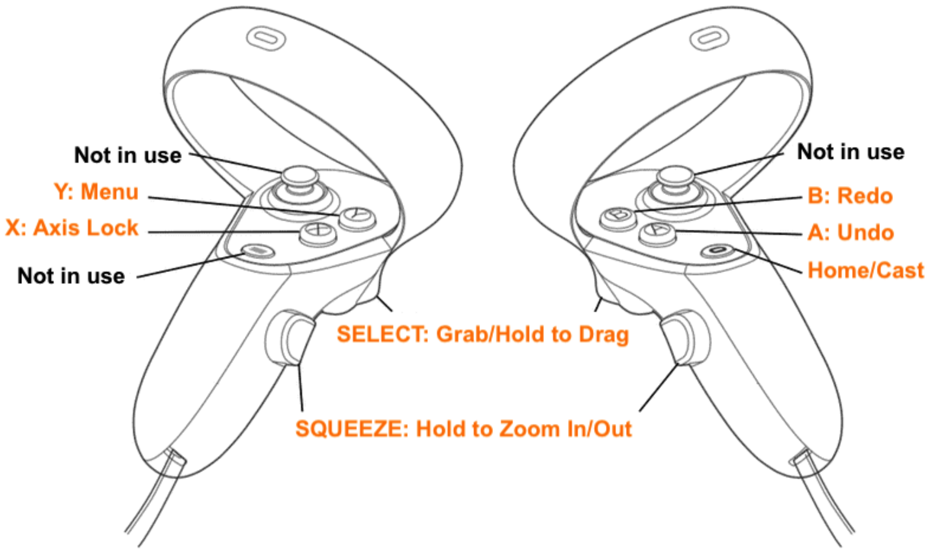
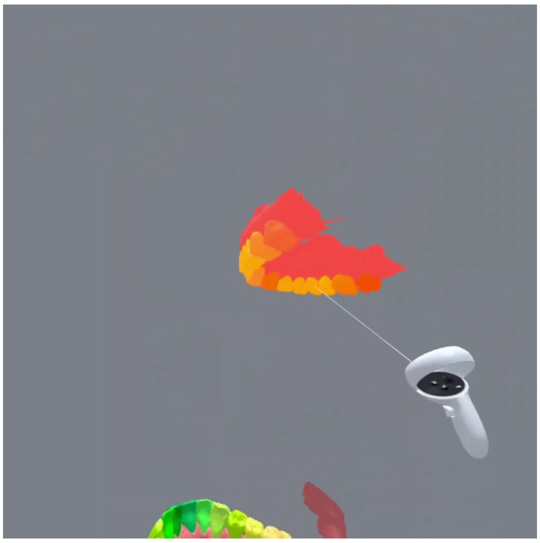
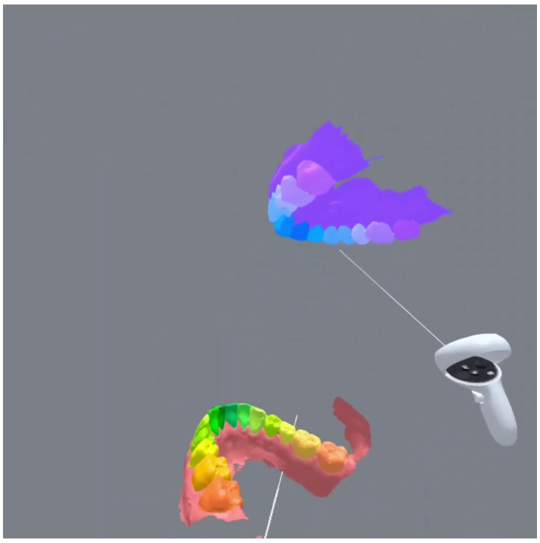


Figure 17: VR controls [2]

The controller casts a ray that indicates the direction it is pointing in. Using this, one can aim the controller at a jaw. When this happens, the jaw gets highlighted in red, indicating it can be moved. By holding the *SELECT* button, one can grab a jaw and move it around. This will highlight the jaw in blue. Releasing the button will deselect the jaw and cause it to stop moving.
Note that one can also grab two jaws at the same time and interact with them in this manner, by using both controllers simultaneously. The collision detection prevents the two jaws from moving into one another, and allows for a realistic sense of movement.



(a) Controller aims at jaw



(b) Controller grabs jaw

For more precise movements, the user can use the *X* button to decide on which axis they want the jaws to move before selecting them. By repeatedly pressing this button, one can cycle through the options: X, Y and Z to help move the jaw on the corresponding axis. Any movement of the controller will then only affect the coordinate of the corresponding axis. Completing the button-pressing cycle will allow the user to once again move the jaw freely in the 3D space.

Additionally one can zoom in and out gradually by pressing and holding the *SQUEEZE* button on the right and left controller respectively, for making edits on a different scale.

There's also an option to undo and redo any changes made by dragging one of the jaws. These actions can be performed with buttons *A* and *B* respectively.
The VR environment will automatically save edits to the scan every so often, so that the progress remains even if the user forgets to save manually. Saving manually can be done through the menu, see Figure 19.
The menu gets triggered by pressing button *Y* and also offers a *Load* option. This will display a list of previous saves, which the user can navigate through with arrows on the screen. Upon selecting a save, the position and orientation of the jaws will then be reset accordingly.

The *Quit* option allows the user to exit the VR environment and will redirect it to a page informing them it's safe to exit the application. The PC of the client will also be notified that the session on the VR headset is ending.

If within one session too many wrong moves have been made the jaws can be reset to their original centre position by selecting *Reset* inside the menu.

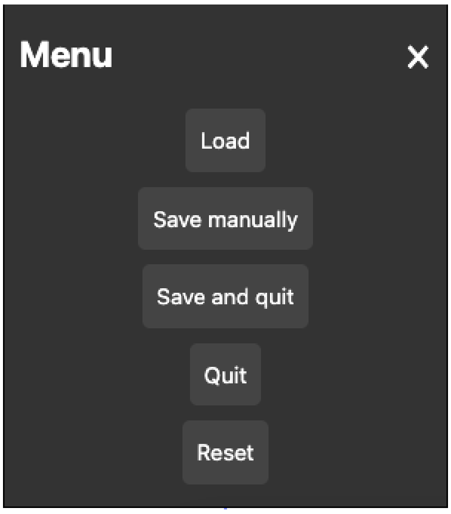


Figure 19: Menu options

7 SWOT analysis — Thomas Speltincx

Strengths	Weaknesses
• The VR environment provides a realistic and immersive experience for dentists to test jaw occlusion, allowing them to see and experience the effects of different bite positions in a virtual environment. • The VR environment can simulate a wide range of jaw movements and occlusion scenarios, allowing for more comprehensive testing and diagnosis than traditional methods. • The VR environment allows for a more time efficient procedure which is also a benefit for the patient. • VR glasses are portable and easy to use, allowing dentists to test jaw occlusion in their own offices rather than referring patients to specialists or imaging centers. • The VR environment can also improve patient engagement and understanding of their occlusion issues, as they can see the impact of different bite positions in a more tangible way than through verbal explanations alone. • The web app itself can be scaled easily. • The VR Hardware already exists and only needs a browser to run the software. • The subscription-based model can provide dental offices access to the latest technology and upgrades, without the need for extra investments or training.	• The use of VR glasses require specialised training and expertise to use effectively, which may limit their adoption by some dentists who are not comfortable with the technology. • VR glasses, especially performant models, can be expensive to purchase and maintain, which may limit their adoption by smaller dental practices or those with limited budgets. • VR glasses may not be suitable for all patients even dentists, as some may experience motion sickness or other adverse effects from the VR experience. • VR glasses cannot completely replace traditional diagnostic methods, such as physical examination and imaging, as they may not provide a complete picture of a patient's occlusion issues. • The use of standard controllers may not be very intuitive for this use case, even though the learning curve is as gentle as possible. • The subscription-based model may not be suitable for all dental practices, particularly those with limited budgets or those that prefer to purchase equipment outright rather than pay for ongoing subscription fees.
Opportunities	**Threats**
• This new method will completely change the current process dentists are used to. • VR technology is rapidly advancing, which may lead to more sophisticated and accurate simulations of jaw occlusion in the future. • The dental sector is also rapidly advancing and already quickly adapting and implementing new technologies. • Checking for occlusion in a VR environment is more environmentally friendly than 3D printing jaw models. • It results in overall lower costs and is easier to manage (no need for printing material and no possible issues with the 3D printer) • The VR environment can be used in dental schools and training programs to teach dentists and students about jaw occlusion and related issues. • The online environment can be used for remote consultations and second opinions, allowing dentists to consult with specialists or colleagues from a distance. • The subscription-based model can provide opportunities for additional services and features, such as technical support, training, or data analytics, which could further enhance the value proposition for dental practices.	• VR technology is still relatively new and unproven in some dental applications, which may limit its adoption by some dentists or patients. Traditional diagnostic methods may still be preferred by some dentists or patients, particularly those who are not comfortable with technology or who prefer a more hands-on approach. • Regulatory issues, such as patient privacy and data security, may pose challenges for the use of VR glasses in dental applications. • Without sufficient interest from dentists or funding from stakeholders, the adoption and implementation of VR glasses in dental practices may be hindered or delayed. • Other technologies may emerge that offer similar or better solutions for testing jaw occlusion, which could compete with or replace VR glasses in the market. • Competition from other subscription-based services or products that offer similar or better solutions for testing jaw occlusion.

8 Business analysis — Dag Malstaf

In this section, an extensive business analysis of the Minimum Viable Product (MVP) developed for the P&O Entrepreneurship project in collaboration with Relu will be conducted. The aim of this analysis is to evaluate the potential of the digital solution in the dental industry and demonstrate its value to potential investors. To achieve this, the lean canvas methodology will be employed, a widely recognised and effective tool for structuring and assessing startup business models. The lean canvas methodology allows for a systematic breakdown of the business plan into nine key components, enabling the identification and addressing of crucial aspects of the venture. With this approach, a clear and compact overview of the business opportunity that the project creates will by provided. The challenges faced by dental professionals in the oral surgery sector will also be assessed.

By examining each of these segments, a thorough and professional business analysis will be created, highlighting the strengths and opportunities presented by the MVP. This analysis aims to demonstrate the economic viability of the project and its potential to create value within the dental industry and for Relu.

8.1 Problem

The dental industry, particularly the oral surgery treatment planning process, is faced with several challenges that hinder efficiency, increase costs, and negatively impact sustainability. This section will discuss the identified problems.

The main issue in the current oral surgery treatment planning process is its inefficiency and high costs. The traditional workflow involves multiple steps, including creating physical 3D models and re-scanning them to determine the optimal fit for the patient. This approach consumes considerable time and resources, leading to increased costs for both dental professionals and patients.

Another problem is the time-consuming nature of the process, which requires multiple steps and the use of physical models made from non-biodegradable plastic materials. This reliance on plastic models raises sustainability concerns and contributes to the environmental burden. The dental industry needs to explore ways to reduce the time and steps involved in the treatment planning process and to address the sustainability issues associated with the use of physical models.

Furthermore, dental professionals often face challenges in adopting new technologies due to the steep learning curves associated with them. The industry must recognise this issue and emphasise the importance of creating solutions that offer a gentle learning curve, enabling dentists and oral surgeons to transition to new technologies more easily. This focus on user experience could foster quicker adoption rates and help dental professionals reap the benefits of innovative solutions.

Lastly, dental specialists typically prefer to have the final say on the details of a procedure. Any potential solutions should acknowledge this requirement and ensure that they do not impede the specialists' ability to make additional adjustments. By providing support without hindering the specialist's autonomy, the dental industry can strike a balance between efficiency and the expert's decision-making authority, ultimately leading to improved patient outcomes.

By identifying these problems, a foundation is laid for the development of solutions that can streamline the treatment planning process, tackle sustainability concerns, and accommodate the unique requirements of dental professionals.

8.2 Solution

The proposed digital solution aims to address the challenges and inefficiencies within the oral surgery treatment planning process comprehensively. The innovative solutions tries to solve the identified problems but at the same time also creates added value in the dental industry by streamlining workflows, reduce costs and improve overall efficiency.

A key component of the digital solution is the ability to determine the optimal fit for oral surgery without the need for physical models. This eliminates the need for creating physical models, and thus reducing costs and time. At the same time the environmental impact associated with traditional treatment planning processes is minimised. The tool thus addresses the primary concerns in the current workflow, making it a more efficient and sustainable alternative.

Recognizing the challenges dental professionals face in adopting new technologies, the digital solution features a user-friendly interface with a gentle learning curve. This ensures that dentists and oral surgeons can seamlessly transition to the new system, reducing the barriers to adoption and increasing the likelihood of widespread acceptance within the industry. To further assist in the adoption process, training sessions utilizing VR technology can be offered to dental professionals. These training sessions will provide a deep understanding of the workflow and functionality of the digital tool, allowing them to fully utilise its potential. Not only do these training sessions facilitate adoption, but they also generate additional revenue for Relu, contributing to the overall financial success of the project. This revenue potential will be discussed in section 8.6.

Finally, the digital solution is hosted in the cloud, making it easily integrated into existing systems and workflows. This cloud-based approach could enable seamless sharing of 3D files between dental professionals. Making the tool cloud hosted would increase efficient communication and collaboration. The ease of integration and sharing capabilities can contribute to a new, streamlined way for doctors to work together and make more informed decisions on patient care.

8.3 Unique value proposition

The unique value proposition of the project lies in its innovative approach to oral surgery treatment planning by leveraging cutting-edge virtual reality (VR) technology. This solution replaces the current inefficient and costly workflow with a streamlined, cost-effective process that benefits both dental professionals and patients. By utilizing VR technology, dentists can experience an immersive and realistic environment, enabling them to test jaw occlusion and visualise the effects of different bite positions in a controlled setting. This offers a level of detail and understanding that surpasses traditional methods, leading to improved patient outcomes.

The portability and ease of use of VR glasses make it possible for dental professionals to test jaw occlusion in their own offices, eliminating the need for referrals to specialists or imaging centers. Moreover, the VR environment facilitates patient engagement and comprehension of their occlusion issues, as it provides a tangible and visual representation of bite positions that cannot be achieved through verbal explanations alone. The scalability of the web app ensures that the service is globally accessible, while the subscription-based model allows dental practices to access the latest technology and upgrades without significant upfront investments. Furthermore, the environmentally friendly nature of the VR environment, which eliminates the need for multiple physical models, and its potential use in dental education and remote consultations, contribute to its unique value proposition. However, it is essential to address potential challenges, such as the need for specialised training, VR equipment costs, and regulatory issues, to ensure the successful adoption of this innovative technology.

8.4 Customer segments

In order to maximise the impact and market reach of the digital solution, it is essential to identify and target the appropriate customer segments within the dental industry. By tailoring the solution to the needs and requirements of each segment, the project can create added value and drive the adoption of the innovative technology. Three customer segments have been identified as primary beneficiaries of the proposed digital solution.

Dental surgeons and oral surgery clinics: The primary customer segment for the digital solution includes dental surgeons and oral surgery clinics that perform complex dental procedures. The digital tool can provide these professionals with a more efficient, cost-effective, and accurate method for treatment planning, thereby reducing the time and resources spent on creating physical models. This can lead to increased productivity, better patient outcomes, and potentially lower costs for both clinics and patients.

Dental laboratories: These entities are responsible for fabricating dental prosthetics, appliances, and other dental products based on the specifications provided by dental professionals. By integrating the digital solution into their workflow, dental laboratories can streamline their processes and reduce the reliance on physical models. This can result in faster turnaround times, increased accuracy, and ultimately, a more competitive offering in the market.

Dental schools and educational institutions: As centers for learning and innovation in the dental industry, dental schools and educational institutions stand to benefit from the adoption of cutting-edge technology. The digital solution can be incorporated into the curriculum, providing students with exposure to the latest advancements in dental treatment planning.

The digital solution offers significant added value to these various customer segments within the dental industry. By targeting these segments and addressing their unique needs, the project can drive the adoption of the technology and ultimately demonstrate its economic viability in the market.

8.5 Channels

To effectively reach, acquire, and communicate with the target customer segments, a strategic mix of marketing and distribution channels will be employed. These channels will ensure that the digital solution is brought to the attention of dental professionals, dental laboratories, educational institutions, and dental insurance companies, fostering adoption and creating added value within the dental industry.

Direct sales to dental practices and oral surgery clinics will serve as a primary distribution channel, ensuring that the digital solution is directly accessible to its target customer segments. By establishing relationships with these entities, the project can foster trust and facilitate the seamless integration of the innovative technology into the existing workflows of dental professionals.

Free trials and workshops from experts will be conducted at dental trade shows and industry events to showcase the innovative technology and its benefits. These demonstrations will offer dental professionals a hands-on experience, allowing them to gain familiarity with the technology and its capabilities. By providing an opportunity for potential customers to directly engage with the digital solution, these trials and workshops will create a compelling case for its adoption.

User referrals and online marketing are essential for raising awareness and driving interest in the digital solution. Targeted advertisements on relevant online platforms will focus on dental professionals and other stakeholders within the dental industry. Additionally, a referral program will be implemented, offering incentives for existing users to recommend the solution to their peers by providing them with platform credits. These credits can be used towards accessing the platform's features on a pay-per-use basis or other services. This strategy caters to customers who may not require the software consistently or those who wish to evaluate the software before committing to a subscription plan. By offering an appealing credit system in the referral program, the platform can encourage user engagement and create a supportive environment for customer retention and conversion to subscription plans, ultimately contributing to the platform's overall growth.

To ensure a diverse reach, more traditional forms of advertising through dental industry-specific newspapers and targeted letters will be employed to connect with established and older dentists. These offline strategies aim to maximise market penetration and foster a broader acceptance of the digital solution across a wider demographic of dental professionals.

8.6 Revenue stream

8.6.1 Pricing structure

In order to generate maximum value for Relu, the proposed business model encompasses four different revenue streams, each designed to address different segments of the market. By diversifying the revenue streams, Relu can effectively address a wide range of customer needs and preferences, while at the same time maximizing the overall value created for the company.

The first revenue stream consists of subscription-based software licensing, which offers tiered pricing for various user categories. This tiered pricing structure includes Basic, Professional, and Enterprise tiers, catering to small clinics, large clinics and educational institutions respectively. This strategic approach enables Relu to accommodate the specific needs and budgets of each customer segment, thus fostering a more extensive and diverse client base. Moreover, the tiered pricing structure incentivises clients to opt for higher tiers, driving increased revenue generation. A summary of the perks and prices for different subscription tiers is presented in Table 1.

Tier	**Basic**	**Professional**	**Enterprise**
Price per month	€250	€750	€2,000
Max Users	5	20	Unlimited
Support	Standard	Priority	Dedicated
Core Functionality	✓	✓	✓
Advanced Features	**-**	✓	✓
Customizable Features	**-**	**-**	✓
API Access	**-**	**-**	✓

Table 1: Perks and prices for different subscription tiers

The second revenue stream is the pay-per-use licensing for individual cases, introduced through a credit system. This model provides flexibility for customers who might not require the software on a regular basis, as well as those who prefer to evaluate the software before committing to a subscription plan. The pricing strategy for this revenue stream ensures that regular users would find the subscription model more economically viable, contributing to a higher conversion rate for subscription plans.

The third revenue stream involves generating additional revenue through training sessions and support services. By offering training sessions for dentists and researchers, Relu can not only ensure the effective adoption of its software but also create an additional income stream. This approach further strengthens the value proposition of Relu's product offering, as clients can confidently adopt the technology with the assurance of ongoing support and assistance.

Lastly, the fourth revenue stream comprises licensing the software to dental software companies, enabling them to integrate Relu's technology into their products. This strategy allows Relu to penetrate a broader market and expand its reach, while simultaneously creating a symbiotic relationship with other industry players. By licensing its software, Relu can benefit from increased exposure and revenue generation without incurring substantial marketing costs.

The combination of these four revenue streams enables Relu to address diverse market segments and customer needs effectively. The strategic pricing and targeted offerings facilitate the maximization of value creation for the company. This comprehensive approach, coupled with continuous monitoring and adaptation of the business model, can potentially lead to sustained growth and increased profitability for Relu in the dental industry. A detailed overview of the pricing structure for each of the four revenue streams can be found in Table 2, which elucidates the pricing tiers for subscriptions, the cost per credit for pay-per-use licensing, fees for training sessions, and licensing fees for dental software companies.

Revenue Stream	**Price**
Subscription-based software licensing (Basic)	€250/month
Subscription-based software licensing (Professional)	€750/month
Subscription-based software licensing (Enterprise)	€2000/month
Pay-per-use licensing (1 credit)	€10
One-day training session	€500/participant
Licensing fee for dental software companies	€10,000/year

Table 2: Pricing structure for revenue streams

8.6.2 Revenue projections

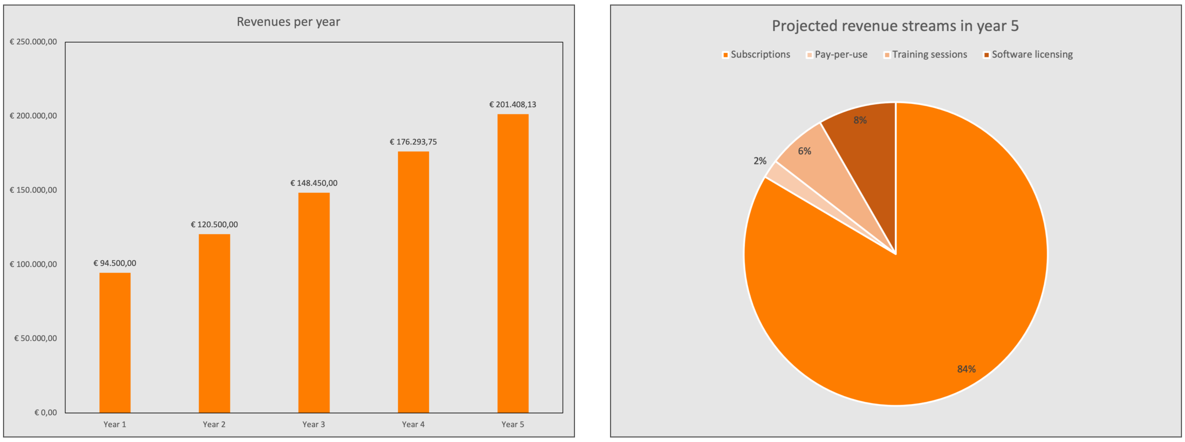
To assess the financial viability and growth potential of Relu's product offering, revenue projections for the first five years of operation have been calculated, taking into account the variable growth rates for each revenue stream as shown in Table 3.

Revenue Stream	Year 1-2	Year 2-3	Year 3-4	Year 4-5
Subscriptions	30%	25%	20%	15%
Pay-per-use	25%	20%	15%	15%
Training sessions	20%	15%	10%	10%
Software licensing	15%	15%	10%	10%

Table 3: Projected year-over-year growth rates for each revenue stream

As illustrated in Figure 20a, a steady increase in total revenue is anticipated throughout the five-year period, reaching €201.408 by the end of year 5. A detailed breakdown of the projected revenue by revenue stream can be observed in Figure 20b. Subscriptions are expected to be the most significant contributor to Relu's revenue, with the Basic, Professional, and Enterprise tiers showing an average terminal growth rate of 15%, as outlined in the table above.

In conclusion, the revenue projections for the first five years of operation demonstrate a promising outlook for Relu's product offering. The diversified revenue streams, strategic pricing, and targeted market approach, combined with variable growth rates, are poised to facilitate sustained growth and increased profitability for the company within the dental industry. Continuous monitoring and adaptation of the business model, alongside a focus on client satisfaction and value creation, will be crucial in ensuring the long-term success and financial stability of Relu.



(a) Total projected revenues for the first 5 years

(b) Revenue diversification in year 5

Figure 20: Revenue projections

Given the reliance on subscriptions as the primary source of revenue, accounting for approximately 85% of total revenue in year 5, a sensitivity analysis was conducted to evaluate the potential impact of changes in the growth rates and subscriptions pricing on the overall revenue projections. This sensitivity analysis of the revenue projections can be viewed in figure 21.

Subscription Revenues in Year 5						Growth Rates				
						5%	10%	**15%**	20%	25%
Basic	Professional	Enterprise	€ 160	€ 480	€ 1.280	€ 76.507	€ 91.080	**€ 107.640**	€ 126.360	€ 147.420
Basic	Professional	Enterprise	€ 200	€ 600	€ 1.600	€ 95.634	€ 113.850	**€ 134.550**	€ 157.950	€ 184.275
Basic	**Professional**	**Enterprise**	**€ 250**	**€ 750**	**€ 2.000**	**€ 119.543**	**€ 142.313**	**€ 168.188**	**€ 197.438**	**€ 230.344**
Basic	Professional	Enterprise	€ 300	€ 900	€ 2.400	€ 143.451	€ 170.775	**€ 201.825**	€ 236.925	€ 276.413
Basic	Professional	Enterprise	€ 360	€ 1.080	€ 2.880	€ 172.141	€ 204.930	**€ 242.190**	€ 284.310	€ 331.695

Figure 21: Sensitivity analysis of projected subscriptions revenues

8.7 Cost structure

The cost structure of the project can be primarily categorised into three major components: cloud hosting costs, sales and marketing and advertisement costs, and customer support and technical maintenance costs. As this project is an extension of the current product offerings of Relu, fixed costs such as insurance and property rent can be ignored.
Cloud hosting costs are essential for maintaining a scalable and accessible online platform that supports the VR solution. These costs are expected to increase as the user base grows and requires more storage and processing power. Sales and marketing and advertisement costs will be incurred for promoting the digital solution and acquiring new customers through online and offline channels, as mentioned in section 8.5. Lastly, customer support and technical maintenance costs will ensure the seamless functioning of the digital solution, timely resolution of customer issues, and continuous improvement of the software.

To align with the revenue projections provided in section 8.6, cost projections for the first five years of operation have been calculated, as shown in Table 4. Assumptions for these costs can be found in appendix B. In Table 5 the assumed growth rates of the costs can be found.

The economic feasibility of the project can be assessed by analyzing the projected gross profit and its potential impact on Relu's financial performance. Over the course of five years, the project is expected to contribute an average of 13% to Relu's current gross profit for the financial year of 2022.

Year	Cloud Hosting	Sales & Marketing	Support & Technical Maintenance	Total
1	€15,000	€15,000	€51,975	**€81,975**
2	€19,500	€19,500	€66,275	**€105,275**
3	€23,400	€23,400	€81,647	**€128,447**
4	€26,910	€28,080	€96,961	**€151,951**
5	€29,870	€32,292	€110,774	**€172,936**

Table 4: Cost Structure over Five Years

Cost Structure	Year 1-2	Year 2-3	Year 3-4	Year 4-5
Cloud Hosting	30%	20%	15%	10%
Sales & Marketing	30%	20%	20%	15%
Support & Technical Maintenance	20%	20%	20%	20%

Table 5: Projected year-over-year growth rates for each cost component

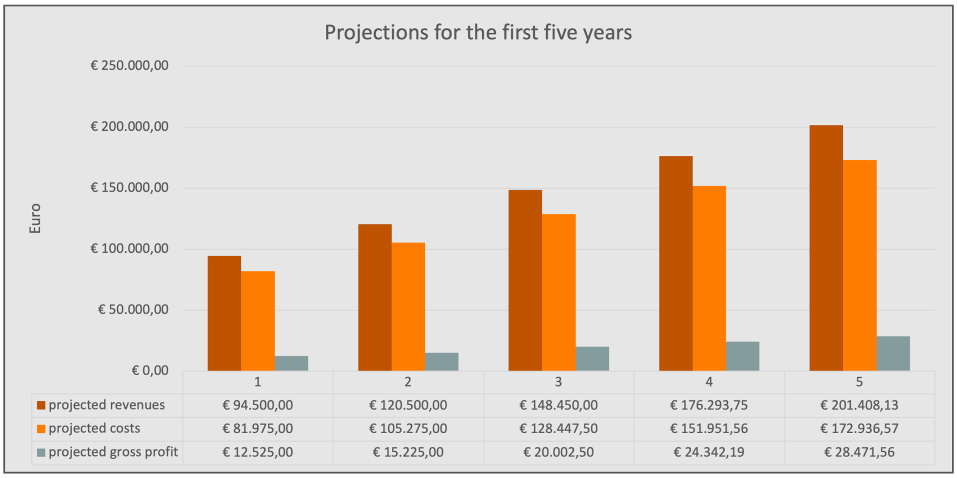


Figure 22: Projected Revenues, Costs, and Gross Profit over Five Years

In the initial stages, the project yields a 12% increase in Relu's gross profit, demonstrating its potential to add value to the company's financial performance. This upward trend is maintained in subsequent years, as the project's impact on Relu's gross profit increases to 13% in the third year. This positive trajectory is sustained throughout the fourth and fifth years. By strategically investing in this digital solution and closely monitoring its progress, Relu can leverage the projected growth in gross profit to strengthen its position within the dental industry. The economic feasibility and growth of the project over the five-year period is visualised in Figure 22, which provides a clear representation of the projected revenues, costs, and gross profit.

These cost projections are based on assumptions which can be found in the appendix B. The actual costs may vary depending on the real-world growth and expansion of the business. A sensitivity analysis for both costs and gross profits has been conducted to account for potential variations in the projections and can be viewed in figure 23 and 24. These projections provide a solid foundation for planning and budgeting purposes, allowing Relu to allocate resources strategically and ensure the long-term success and financial stability of the digital solution within the dental industry.

Total Costs in Year 5		Growth Rates				
		10%	15%	**20%**	25%	30%
Total	€ 9.600	€ 83.256	€ 92.170	**€ 101.871**	€ 112.409	€ 123.837
Total	€ 12.000	€ 104.069	€ 115.213	**€ 127.339**	€ 140.511	€ 154.797
Total	**€ 15.000**	**€ 130.087**	**€ 144.016**	**€ 159.173**	**€ 175.639**	**€ 193.496**
Total	€ 18.000	€ 156.104	€ 172.819	**€ 191.008**	€ 210.767	€ 232.195
Total	€ 21.600	€ 187.325	€ 207.383	**€ 229.210**	€ 252.920	€ 278.634

Figure 23: Sensitivity analysis of projected total costs

Projected profits in Year 5						Growth Rates				
						5%	10%	**15%**	20%	25%
Basic	Professional	Enterprise	€ 160	€ 480	€ 1.280	€ 26.472	€ 32.130	**€ 38.990**	€ 47.172	€ 56.803
Basic	Professional	Enterprise	€ 200	€ 600	€ 1.600	€ 24.785	€ 31.858	**€ 40.432**	€ 50.659	€ 62.699
Basic	**Professional**	**Enterprise**	**€ 250**	**€ 750**	**€ 2.000**	**€ 22.676**	**€ 31.517**	**€ 42.235**	**€ 55.019**	**€ 70.069**
Basic	Professional	Enterprise	€ 300	€ 900	€ 2.400	€ 20.567	€ 31.176	**€ 44.037**	€ 59.379	€ 77.438
Basic	Professional	Enterprise	€ 360	€ 1.080	€ 2.880	€ 18.037	€ 30.767	**€ 46.201**	€ 64.610	€ 86.282

Figure 24: Sensitivity analysis of the projected gross profits

8.8 Key metrics

To effectively assess the progress and success of the project, it is essential to establish and monitor key metrics that align with the overall business objectives. These key performance indicators will provide valuable insights into the areas where the product is excelling and those that require further attention or improvement. The following key metrics have been identified for evaluating the performance and impact of the VR tool:

Reduction in treatment planning time and cost: One of the primary objectives of the product is to streamline the treatment planning process and minimise costs for dental professionals. By measuring the reduction in time and cost, it will be possible to quantify the efficiency gains achieved through the implementation of the VR solution.
Number of exported scans used in medical procedures: This metric highlights the practical application of the VR solution in the dental industry. By quantifying the number of exported scans used in medical procedures, it will be possible to evaluate the extent to which the solution is impacting patient outcomes and overall dental practice efficiency.

Adoption rates among new users: Monitoring the rate of adoption by new users is crucial in determining the market penetration and acceptance of the product. High adoption rates are indicative of a successful marketing strategy and a solution that effectively addresses the needs of dental professionals.

Number of referrals: Tracking the number of user referrals will provide insights into the effectiveness of the referral program and the overall satisfaction of existing customers. A high referral rate reflects a strong network effect and increased customer trust in the platform.

Ratio of subscription extension after one year: Measuring the renewal rate of subscriptions is a key indicator of customer satisfaction and loyalty. A high subscription extension ratio suggests that customers find value in the product and are willing to continue investing in the platform.

By regularly tracking these key metrics, the project's progress and success can be effectively evaluated, allowing for timely adjustments to the business model and strategies as needed. A focus on meeting or exceeding these performance indicators will contribute to the long-term growth and sustainability of the digital solution within the dental industry, ultimately reinforcing Relu's unique value proposition.

8.9 Unfair advantage

In a highly competitive market, businesses must leverage their unique strengths and advantages to stay ahead of the competition. Relu's digital solution possesses several unfair advantages that are difficult for competitors to replicate, ultimately contributing to the company's long-term success within the dental industry.

One of the key unfair advantages of Relu's digital software is its proprietary AI-driven 3D scan segmentation technology. This innovative artificial intelligence technology has been specifically designed for 3D scan segmentation, offering a significant competitive edge over other solutions in the market. The AI technology can be easily integrated into the VR tool, creating a unique product that provides more accurate and efficient scan processing, leading to improved treatment planning and patient outcomes. By continuously developing and refining this technology, Relu can maintain its competitive edge in the dental industry.

Another critical advantage is the first-mover advantage in the market. Being among the first to introduce a comprehensive VR-based dental treatment planning solution, Relu positions itself as a pioneer in the field, allowing the company to establish a strong brand presence and customer base before competitors enter the market. This advantage enables Relu to benefit from valuable feedback from early adopters and refine the digital solution to better cater to the evolving needs of dental professionals.

Lastly, the product offers a complete integrated solution and platform that seamlessly connects with other services offered by the company. This comprehensive and cohesive approach provides dental professionals with a streamlined and efficient workflow, thereby improving their overall experience and satisfaction with the platform. The integration of multiple services under one unified platform creates a holistic offering that is difficult for competitors to match.

By capitalizing on these advantages and continuously striving for innovation, Relu can reinforce its unique value proposition and solidify its position as a leader in the dental industry.

9 Stakeholder opinion — Astrid Dewever

9.1 Introduction and stakeholder group identification

The feedback and opinions of stakeholders, who have a vested interest in the success of a software project, are invaluable in ensuring that the project meets their needs and expectations, which can improve the user experience, increase adoption rates, and ultimately contribute to the success of the project. The aim of this section is to evaluate the feedback gathered from each stakeholder group and determine its impact on the final outcome of the project. Throughout the different subsections, valuable insights can be gained into the project's strengths and weaknesses, opportunities for improvement can be identified, and recommendations can be developed to optimise the platform's functionality and user experience.

The stakeholders consist of the company Relu, who commissioned the project and the users of the new technology, being professionals in the dentistry world. The patients themselves are not considered stakeholders as they do not determine the planning of a maxillofacial surgical procedure and therefore do not have a say in the workflow of the doctor. However, it is appropriate to acknowledge that it is generally more pleasant for the patient not to have to undergo alginate impression taking, as this often induces a gag reflex and may lead to a level of discomfort.

During the project's initial phase in the first semester, a dedicated group of students gathered feedback from the company Relu and dental professionals in parallel with the development of the four different minimum viable product (MVP) proposals. This feedback informed the project's vision and helped the team identify key priorities, contributing to the successful implementation of the VR functionality in the final product.

9.2 Methodology

9.2.1 Methodology to reach Relu as stakeholder

In order to obtain input on the project by Relu itself, two Zoom meetings were held during which discussions took place on the design process itself, as well as feedback from their side regarding key components in the design of the application. Further information of what has been discussed during these two Zoom meetings can be read in the Stakeholder opinions section.

9.2.2 Methodology to reach professionals in the dentistry field as stakeholders

A survey was conducted using Google Forms to gather feedback from professionals in the dental industry, regarding the development of VR technology and its potential to improve workflow. The survey consisted of several questions, beginning with inquiries into the respondents' level of familiarity with VR in order to assess the degree to which this technology has been adopted in the industry. The survey also sought to discover respondents' preferences and other factors that may impact the design process of the VR application.
The survey was disseminated through various channels to enhance its reach and obtain a higher response rate. Initially, it was distributed to orthodontists and other individuals known to the group members. Additionally, the survey was posted on the official channels of relevant associations, including the "Verbond Vlaamse tandartsen " (VVT) and the "Leuvense Universitaire Tandheelkundige Vereniging" (LUTV). Following correspondence with the chairman of LUTV, Em. prof. dr. Antoon De Laat, the survey was subsequently published on their website (accessible at https://www.lutv.be/).

The rationale behind the chosen method of gathering opinions of the concerning stakeholders is essential to explain the actions taken by the team and justify the choices that were made. The decision to conduct the meetings with the company via Zoom was based on the team's opinion that it allowed for a more personal and dynamic interaction with the stakeholders, while also being time-efficient given their busy schedules.

The decision to choose the online survey method was based on its time efficiency, wide reach, structured data and quick analysis. Through targeted email outreach to known specialists and posting on relevant channels of associated organizations, the team was able to achieve direct access to the intended target audience. However, there were also some disadvantages to this method, including the potential for a low response rate and a limited personal interaction. Despite these concerns, it is important to reach solely highly educated peers in the dentistry business to gain insights into the development of VR in this field. Providing an open-ended section in the survey for respondents to provide further input and comments allowed for a more nuanced understanding of the stakeholders' viewpoints and their diverse perspectives. This approach enabled the team to gather valuable insights that would have otherwise been missed in a closed-ended survey format, thus enhancing the overall quality and richness of the gathered intelligence. Overall, it is no question that the benefits of using an online survey outweighed the drawbacks and provided valuable data to inform the team's research and project development.

Despite the effort to reach as many stakeholders as possible, eventually a small dozen responses were received. Thanks to the open structure of the survey, these were all of a high-quality nature with a great deal of personal input, but the sample size remained small. The limitations of the survey include a small sample size, a relatively low response rate, and a niche group of the general population. Thus, the inferences that can be drawn from the survey's findings are constrained, and it is crucial to acknowledge the study's limitations when drawing conclusions.

9.3 Stakeholder opinions

9.3.1 Opinion of Relu

The design of the application focused on the user workflow and user's experience, which Relu also considered a good approach. They also emphasised that users should be able to easily transition from the 2D web application to the 3D VR world in a smooth and intuitive way, without the need for multiple clicks or unneeded complicated steps. This was taken into consideration for the final product (see section 6.1). From the dashboard where various scans of one patient are displayed, it is possible to go to the VR application in one click by clicking on a large icon. In addition, it is possible to first open a scan and display the 3D model rotating in the browser page before going to the VR application from the patient's dashboard. The latter method is useful if the expert simply wants to view the scan in full compared to a smaller figure in the dashboard, possibly before they open it in VR.

To facilitate a smooth transition to VR, Relu gave the advice to use Three.js, a JavaScript library, to build the VR application. They expressed concerns about the possible use of Unreal Engine, which is designed for building heavy applications such as high-graphic games. This concern, along with the fact that programming in Unreal Engine requires a laptop with a graphics card, which only two members of the team possessed, it was decided not to use this library.

The second topic that Relu explicitly spoke about is where to perform the intensive computation related to the VR environment. Currently, everything is done on the VR headset computer itself, and there is no requirement for the computing power of the user's device. Relu imposes a minimum requirement of a 4GB video RAM GPU on its users, which thus far is not necessary for the developed application. The performance of the computation in the VR headset is increased by the use of web workers, where the respective JavaScript runs independently in the background while other JavaScript on the HTML page runs.

Next, Relu also advised the team to definitely make use of decimation. Decimation refers to the process of reducing the number of polygons in a model's mesh in order to decrease its size. This results in a smaller 3D model file size, as well as reducing the workload that the GPU needs to render the model, which in turn improves performance. The advice was followed and decimation was incorporated into the project. For more information, please refer to section 5.4.

9.3.2 Opinion of professionals in the dentistry field

The survey was predominantly completed by dentists and orthodontists, with the second largest group consisting of periodontists. (*Periodontology* is a medical specialty within dentistry that focuses on the supporting structures of the teeth and molars: the jawbone, the gums, and the periodontal ligament.)

Previous experience with VR technology Initially, the participants were asked about their level of experience with VR technology in their professional and personal domains, indicating their degree of acquaintance with this novel innovation. It was found that 55% of the respondents lacked prior familiarity with VR, while the remaining respondents reported previous exposure, primarily in commercial gaming. Notably, one respondent reported using VR as a tool to alleviate anxiety, when treating young patients to put them at ease when performing dental procedures.

Potential use of VR in their profession The responses to the question about the potential use of VR in the specialist's work can be categorised into three main purposes, with respondents seeing potential in virtual planning, guidance and visualization during surgery, and visual representation of the end result.

The first purpose, virtual planning, was suggested in a broader sense than what is being employed in the project, being proposed for both implant surgeries and aesthetic surgeries, besides maxillofacial surgery as intended by Relu.

The second purpose of VR serving as visual guidance during the surgery included walking the patient through the entire procedure or operation, with supporting visualization of the various steps.

The third purpose raised was the possibility of using VR to show the patient what the result of the treatment could be. Multiple responses also suggested that it can be used for aesthetic purposes to display the end result. Additionally, it was suggested to use VR for a composite wax-up, where restorations are planned on a plaster model to complete repairs to the teeth.

Important features This question aimed to assess the key characteristics of the product that needed to be taken into account during the design process. A total of six features were provided, with the respondent being asked to rank them based on personal interest and importance. The average ranking of every feature was then be calculated, giving the following result (with $x < y$ meaning that x was ranked less important than y): cost/cost model < small learning curve < layout < closeness to reality < precision. The final project's design process took into account the features that were indicated to be the most important, namely as precision, closeness to reality, and layout.

Other remarks Lastly, during the survey, professionals were given the opportunity to express any concerns or comments. It was rightfully pointed out that there might be compatibility issues with the combination of the VR headset and dental surgical loupes, which are magnifying glasses commonly used for dental surgeries. This issue could indeed pose a problem, particularly if the VR environment is also used to assist during the operations. However, Relu did not seem to be concerned about this, and thus no modifications of the design or project in general were made to accommodate the matter.

9.4 Analysis and conclusions

From the side of the first stakeholder group, being Relu, there were no strict restrictions regarding the design process that came out of the several Zoom calls. However, emphasis was placed on several design aspects, such as the workflow on the website needing to be smooth and clear for the user. An easy transition to the VR environment was crucial. They also stressed the importance of determining where the intensive calculation of the VR environment should take place, whether in the VR headset, on the user's computer, or in the cloud, as this could be a hurdle for the user. The product had to be compatible with the user's laptop, where a minimum of 4GB video RAM GPU could be required. These considerations were taken into account, and the developed design imposes no such obligations on the user's device.

From the other stakeholder group, the dentists (the actual end-users of the platform), suggestions were taken into account regarding potentially useful ways of using the VR technology and what the most important features are. There are future opportunities to expand the use of VR from virtual planning to more broad possibilities, like guidance and visualization during surgery, as well as a visual representation of the end result. This last application is present in the final product, as the user can save the relative position of the jaws and view the transition from beginning to result accordingly. The three most important features taken into account during the design process are precision, closeness to reality, and layout. Less important features for dental professionals were found to be cost/cost model and a small learning curve and were granted less importance.

10 Scalability analysis — Astrid Dewever

As the product gains traction and more users join over time, it is essential to ensure that it can scale efficiently. Scaling refers to the ability of the product to accommodate increasing demand for resources, such as storage space and computing power, as the number of users and associated data grows. Failure to scale appropriately can lead to lower performance and reliability, and ultimately, a loss of customers. Therefore, conducting a scalability analysis is necessary to assess the developed product's ability to handle growing demand and identify potential bottlenecks that may arise as it scales in the future.
To already give away part of the answer, cloud computing offers a flexible and cost-effective solution for managing large volumes of data and users. By making use of cloud resources, the product can easily scale, without the need for significant upfront investments in hardware or other infrastructure. In this section, the scalability of the developed VR product and the benefits of using cloud computing will be discussed.

10.1 Current infrastructure and limitations

The project uses a Meta Quest 2 VR headset with two controllers and a web page to host the application that waits for the user to press a button on their computer. Once the button is pressed, a scan of a patient's jaws can be observed through the VR headset. The VR glasses and desktops of all customers connect to the server via Go channels, where the communication is implemented using Golang with protocol buffers. An automatically generated code is used as authentication instead of a login on the glasses to ease the customer's usage, and the final position of the jaws is stored in the database after the scan is complete.

The VR environment is implemented using Three.js and Cannon.js for visualization and interaction of the jaws respectively. The jaw meshes, controller meshes, and targets are the relevant Three.js data structures used. The application is implemented as a PWA, as it is listed as a VR app in the application list, making it easier to open, and easier to implement than a native Meta Quest 2 app. The PWA was created using Bubblewrap CLI.

The VR application hosted on a web page was chosen due to several reasons. First, it is easier to implement and faster to redirect in a browser or PWA than a native application. Second, a PWA is listed as a VR app, making it easier to open than entering a link in a browser. Third, a PWA is easier to implement than a native Meta Quest 2 app.

The VR environment provides features such as 360-degree view, jaw movement, collision handling, haptic feedback, axis locking, object locking, zooming, saving, before/after animation, reset, and casting. The collision handling and jaw movement are the two core features of the VR environment. The jaws can be moved around with the VR controllers, but the software ensures that both parts of the teeth cannot overlap in the three-dimensional rendered VR space. If the upper jaw touches the lower jaw in the simulation, the controllers will vibrate to indicate that further movement is not possible.

The authentication mechanism implemented using an automatically generated code instead of a login on the glasses may not be secure enough for a complete product. The PWA was chosen due to its compatibility with the Meta Quest 2 headset, but it may not be the best option for other VR headsets. Furthermore, the VR environment is limited to rendering scans of patients' jaws and manipulating them. Potential future features such as rendering and manipulating other parts of the body or incorporating machine learning may be out of scope for this project.

10.2 Scaling options and considerations

When it comes to scaling an application, there are three options to consider. The choice of scaling option largely depends on the specific needs and requirements of the application. The first two of the most common scaling options are vertical scaling and horizontal scaling.

Vertical scaling involves increasing the resources allocated to a single server that is used, such as adding more RAM, CPU, or storage capacity. This approach is generally easier to implement and can be more cost-effective for smaller applications. However, there are limitations to how much a single server can be scaled vertically, and once those limits are reached, the only option to further scale is horizontally.

Horizontal scaling, on the other hand, involves adding more servers to the application's infrastructure, allowing for greater resource availability and load distribution. In classic thinking, horizontal scaling is the first thing that comes to mind. However, this approach is more complex and requires more planning and management, but it allows for virtually unlimited scalability, which is an advantage. Horizontal scaling is often used for larger applications that demand high availability and fault tolerance as well.

When deciding which scaling option to choose, it's important to consider factors such as the architecture of the application, the patterns in user traffic, and cost constraints. For example, if the VR application has a high volume of user requests that can be handled by a single server, vertical scaling may be sufficient. However, if the VR application requires specifically a high availability and fault tolerance, horizontal scaling may be necessary as well.

A second consideration is the use of cloud infrastructure and services. Cloud providers offer a variety of scaling options and services, such as auto-scaling, which automatically adjusts the number of servers based on demand. This can be a cost-effective and efficient way to handle scaling, but it requires careful monitoring and management to ensure optimal performance and cost efficiency. The decision to use Google Cloud for the VR project was based on a short analysis of available options. Google Cloud was deemed to be suitable for the overall project, providing a robust and scalable infrastructure at a reasonable cost. Alternative cloud computing platforms that offer comparable services are Amazon Web Services (AWS), Microsoft Azure and Oracle Cloud, but have not been chosen.

11 Future prospects — Astrid Dewever

As the dental industry advances and integrates new technologies, there exist ample prospects for the development and improvement of digital platforms for dental professionals. With the successful implementation of the current platform, it is prudent to explore potential future enhancements and expansions that could further augment the platform's capabilities and advance patient outcomes. This section aims to discuss potential future prospects for the platform, examining opportunities for expansion, novel features, and technological advancements that could propel the platform forward in the coming years.

11.1 In general

Mobile compatibility To improve accessibility and convenience for dental professionals, a mobile version of the platform could be developed, enabling on-the-go access to patient scans and notes. This would allow for increased flexibility and mobility in managing patient information, especially in situations where a desktop or laptop computer is not readily available.

Enhanced collaboration and communication features To improve communication and collaboration among dental professionals, it may be beneficial to integrate real-time communication and collaboration features into the platform, such as video conferencing or messaging. This would enable more efficient and effective collaboration between dental professionals, potentially leading to improved patient outcomes and better overall care.

Integrating with hospital programs The platform could provide a link to other hospital programs, such as the *Klinisch Werkstation (KWS)*, a tool by nexuzhealth. KWS is a platform that allows healthcare facilities to access centralised patient data for optimised patient care. By linking to KWS, the platform could potentially streamline the access and sharing of patient data, facilitating more efficient and effective treatment.

Expansion to other medical specialties To expand the reach and impact of the platform, it may be worth considering opportunities to branch out into other specialties beyond dental professionals, such as orthopedics or neurology, that could benefit from similar scanning and visualization capabilities. This would allow for greater utilization of the technology and could potentially increase revenue streams for the company.

Provide tutorials or training programs To accelerate the learning curve for dental professionals who are new to the platform, tutorials or training programs could be provided. These resources would aim to help users quickly become proficient in using the platform's various features, reducing the time needed for onboarding and enabling them to provide high-quality care more efficiently. The implementation of such tutorials or learning programs aligns with the stakeholder feedback provided by dental professionals, who identified the importance of reducing the learning curve as a key area for improvement.

11.2 The VR environment

Customization to the user's preferences In terms of improving the VR functionality, it may be worthwhile to explore additional features such as the ability to annotate or mark specific areas of the scan, as well as integration with other tools commonly used in dental practices. Additionally, it could be useful to add more user preferences such as different lighting options or the ability to customise the color of the scan to the VR environment.

Casting movements in VR to a desktop By implementing a more advanced casting functionality, the VR environment could be enhanced with the ability to share the VR experience with patients or other stakeholders who are not wearing the VR goggles. This would allow them to follow along with the actions of the doctor and gain a better understanding of the treatment plan. In turn, this could increase patient satisfaction and trust in their dental care.

Support for haptic feedback gloves In order to enhance the VR experience for dental professionals using this platform, the integration of haptic feedback gloves might be beneficial. By incorporating these gloves, users would be able to physically interact with the virtual environment and receive tactile feedback, allowing for a more immersive experience. This could also improve the accuracy of tooth movement in the VR environment, leading to more precise treatment planning and outcomes as indicated most important by the stakeholders opinion.

Improve collision detection performance To enhance the user experience and make the simulation feel more natural, it may be worth exploring ways to improve the performance of the collision handling in the VR environment. One solution could be to implement cloud computing to offload the processing power required for collision detection and response, resulting in a more seamless and responsive simulation. This would create a more captivating and authentic experience for users, increasing their sense of presence within the virtual environment.

Adding a heat map One way to improve the VR environment is by implementing a heat map to visualise areas of the scan that require further attention or are particularly problematic. This would provide a more efficient and effective way for dental professionals to identify issues and develop treatment plans. Additionally, the heat map could be used as a tool for patient education, allowing them to better understand their dental health and the specific areas that require attention.

Automated analysis and diagnosis To enhance the platform's capabilities, there is potential to incorporate optimization and machine learning algorithms to analyse scans and provide automated diagnoses or treatment recommendations. This could potentially improve the accuracy and efficiency of diagnosis, and could ultimately lead to a better treatment for patients.

11.3 The server-side implementation

Security and data encryption To ensure the security and privacy of patient information, it is imperative that the platform implements appropriate measures for user authentication and data encryption. Currently, the login process does not provide encryption and user data is stored in the back end database. However, if the developed product were integrated with an existing platform, this concern would be alleviated and the existing authentication system could be adopted.

Transition to Kubernetes Implementing Kubernetes can improve the scalability and reliability of the VR tool by automating the deployment, scaling, and management of the application across multiple nodes. This can help ensure that the application is always available and can handle increasing amounts of traffic or usage. Additionally, Kubernetes can provide features such as load balancing and rolling updates, which can further enhance the user experience by improving the responsiveness and stability of the application.

11.4 Recommendation

To conclude, it is important to continue to gather feedback from early adopters and incorporate it into the development process to ensure that the platform continues to meet the needs of its users.

12 Conclusion — Jonas De Schouwer

In the context of P&O Entrepreneurship, a team of nine computer science students was assembled to undertake a project commissioned by Relu, a company specialised in software for teledentistry. The objective was to improve the workflow for oral surgery procedures in terms of efficiency, cost-effectiveness and sustainability.

Many possible solutions were conceived and eventually four were selected and converted into minimal viable products. After collecting feedback from the main stakeholders, being Relu and the potential users, one design was chosen for further development.

The proposed solution is a web application that enables dentists to manage intraoral scans and view them in a virtual reality environment. The web app features a login system, a patient overview, a scans dashboard and several additional features that were implemented to enhance the user experience. These features include tags, filters, a search bar, and various import/export functionalities. The VR environment is the centerpiece of the solution, as it allows dental professionals to find a beautiful occlusion for the patient's jaws without requiring a physical model. Additionally, it can serve as an effective tool to educate patients and students on the procedure at hand. The environment is loaded with features that ensure a seamless user experience. First of all, it is possible to walk around the jaws to get a full picture of the situation. Second, a user can grab and move the jaws using the VR controllers, while the collision handling software makes sure that no physically impossible scenarios occur. Many more features were implemented, for a full list please refer to section 5.4.1. Both the web app and the VR environment are connected to a database that is hosted on Firebase. This database houses the scans and other related support files.

This solution was established in many steps. In the first step, various technologies were examined and the most suitable ones were added to the project's tech stack. Second, the team was divided into four dynamic subteams, each responsible for one aspect of the final project. The respective aspects were the web app, the back end, the VR environment and the integration of VR within the web app. Next, the project itself was broken down into tasks, which were grouped as sprints according to an Agile Development methodology. The team was supported in this by the supervision team, who provided a workshop on Agile Development and offered some helpful tips and tricks. Lastly, the application needed to be implemented. This phase occupied the entire second semester, with work being done during the programming sessions (one or two each week) and at home. Every Tuesday coding session began with a stand-up meeting in which every subteam communicated their progress and struggles. Furthermore, there was a larger internal presentation at the end of each sprint. This turned out to be a good idea, as it kept all team members informed of the progress of the overall project and provided some extra motivation to meet intermediate deadlines.

At all stages of development the objective of creating a profitable product was taken into account. A thorough analysis of this was made and discussed comprehensively in section 8, which describes how the created software can be taken to market. Concretely, a combination of revenue streams coming from four main customer segments will enable Relu to address diverse market segments and customer needs effectively. Both revenues and costs were forecast and aggregated in a profit projection, which looks promising for Relu.

Before the result is taken to market, however, some further enhancements would be beneficial. The most crucial improvement would be a performance boost to the collision handling algorithm to ensure a natural sensation. Secondarily, additional features may be added to the web app and the VR environment. Some of those that would be useful are the integration of (real-time) communication mechanisms in the web app, a heat map in the VR environment, a way to cast the VR view to the desktop, and an optimization algorithm that makes an occlusion proposal automatically. Of course, collecting feedback from early adopters and incorporating it into the development process remains essential to ensure that the platform continues to meet the needs of its users.

In conclusion, all qualitative goals that were set at the start of the project were achieved. A fully integrated software product was created that is natural to work with and enhances the workflow of oral surgery operations. Moreover, the product utilises cutting-edge VR technology and still has a lot of potential for future innovations. Finally, the business analysis shows that the created software can be made profitable, albeit requiring some further development.

The project was definitely an impactful educational experience for the entire team. For most students, this was the first functional and fully integrated application they ever created, so both the large scope of the project and the size of the team were challenges that needed to be overcome. Both were successfully tackled using an Agile Development methodology, the workshop offered by the supervision team proved instrumental in this regard. Of course, the team members also gained extra individual programming experience, in particular with languages and frameworks that had not been previously studied at university, like JavaScript, Go, Docker, React, and Three.js. Furthermore, it was instructive to conduct an analysis of the economic value of the result ourselves, taking inspiration from the lean canvas procedure. We would like to express our gratitude to Relu, professor Berbers, and the didactic team for offering us this opportunity and supporting us throughout the project. The skills and knowledge acquired will undoubtedly have a lasting impact on our future academic or professional careers.

13 Appendix

A GitHub repository

The entire source code of this project is hosted on GitHub and freely accessible via the following link: `https://github.com/LarsDepuydt/peno-entrepreneurship-3d-oclusion`.

B Cost assumptions

Cost Assumptions		
Cloud Hosting Costs	€50/two weeks * 26weeks * 10 for scale + 2000 for buffer	
Sales & Marketing Costs	Arbitrary choosen, can be modified based on growth	
Support & Technical Maintenance	Relu has 55% expense margin for loans social security charges and pensions	see 2022 financial statements

Figure 25: Assumptions for cost projecting

References

[1] "Db diagram." https://dbdiagram.io/home. Accessed: 2023-05-05.

[2] "Oculus quest unity controller buttons." https://blog.nobreakpoints.com/oculus-quest-unity-controller-buttons-verwenden/. Accessed: 2023-05-08.

[3] "What is scrum and how to get started." https://www.atlassian.com/agile/scrum. Accessed: 2023-05-05.

[4] "Jira." https://jira.atlassian.com/. Accessed: 2023-02-12.

[5] Y. Bai, "Best practices: Why use golang for your project." https://www.uptech.team/blog/why-use-golang-for-your-project. Accessed: 2023-05-02.

[6] "Github language stats." https://madnight.github.io/githut/#/pull_requests/2023/1. Accessed: 2023-05-03.

[7] M. Narang, "Why use docker? advantages and tools." https://www.knowledgehut.com/blog/devops/why-use-docker, 2023. Accessed: 2023-05-04.

[8] B. Simon, "Three.js journey - physics." https://threejs-journey.com/lessons/physics#ammo-js, 2023. Accessed: 2023-05-06.

[9] "Connect docs introduction." https://connect.build/docs/introduction/. Accessed: 2023-05-05.

[10] "Protocol buffers documentation." https://protobuf.dev/. Accessed: 2023-05-05.

[11] "Buf." https://buf.build/. Accessed: 2023-05-05.

[12] react@18.2.0 documentation, "useeffect - react." https://react.dev/reference/react/useEffect, 2023. Accessed: 2023-05-06.

[13] K. Mamou and F. Ghorbel, "A simple and efficient approach for 3d mesh approximate convex decomposition," pp. 3501–3504, 11 2009.