FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGY CENTER
AUTOMATION AND SYSTEMS DEPARTMENT
UNDERGRADUATE COURSE IN CONTROL AND AUTOMATION ENGINEERING

Fernando Bandeira de Mello Mattos

Development of an Application on a Digital Twin Platform to Optimize Cutting Tool Usage in CNC Machining

Aachen
2023

Fernando Bandeira de Mello Mattos

Development of an Application on a Digital Twin Platform to Optimize Cutting Tool Usage in CNC Machining

Final report of the subject DAS5511 (Course Final Project) as a Concluding Dissertation of the Undergraduate Course in Control and Automation Engineering of the Federal University of Santa Catarina.
Supervisor: Prof. João Carlos Espíndola Ferreira, Dr.
Co-supervisor: Tommy Venek, Eng.

Aachen
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Mattos, Fernando Bandeira de Mello
Development of an Application on a Digital Twin Platform to Optimize Cutting Tool Usage in CNC Machining / Fernando Bandeira de Mello Mattos ; orientador, João Carlos Espíndola Ferreira, coorientador, Tommy Venek, 2023.
62 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Monitoramento de fresas. 3. Desenvolvimento de software. 4. Gêmeos digitais. I. Ferreira, João Carlos Espíndola. II. Venek, Tommy. III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. IV. Título.

Fernando Bandeira de Mello Mattos

Development of an Application on a Digital Twin Platform to Optimize Cutting Tool Usage in CNC Machining

This dissertation was evaluated in the context of the subject DAS5511 (Course Final Project) and approved in its final form by the Undergraduate Course in Control and Automation Engineering

Florianópolis, December 1st, 2023.

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Course Coordinator

Examining Board:

Prof. João Carlos Espíndola Ferreira, Dr.
Advisor
UFSC/CTC/DAS

Tommy Venek, M. Sc.
Supervisor
gemineers GmbH

Prof. Rodrigo Lange, Dr.
Evaluator
IFRS

Prof. Eduardo Camponogara, Dr.
Board President
UFSC/CTC/DAS

ACKNOWLEDGEMENTS

I want to express my gratitude, first and foremost, to my parents, Mauro and Roberta, for always believing in me and providing support and structure for me to receive a quality education. Without you, none of this would have been possible.

I thank my brothers, Bernardo and Frederico, for being examples to me since the beginning and for giving me good advice, such as encouraging me to pursue an exchange program abroad.

To Professor João Carlos Espíndola Ferreira, my academic advisor, thank you for the valuable insights and feedback given, and for guiding me through this project.

To Tommy Venek, my local supervisor, Vincent Gerretz, and Sven Schiller, thank you for all the help, guidance, and mentoring during my work, and I appreciate gemineers GmbH for the opportunity of this internship and for all the learning I gained during this year.

To all my friends and colleagues at work, especially Victor, Ícaro, Giovanna, Maurici, Pedro, Milena, and Luiz, thank you for making workdays more enjoyable and for helping me every time I needed it.

"The science of today is the technology of tomorrow."
— Edward Teller (1908 - 2003)

DISCLAIMER

Aachen, December 1st, 2023.

As representative of the gemineers GmbH in which the present work was carried out, I declare this document to be exempt from any confidential or sensitive content regarding intellectual property, that may keep it from being published by the Federal University of Santa Catarina (UFSC) to the general public, including its online availability in the Institutional Repository of the University Library (BU). Furthermore, I attest knowledge of the obligation by the author, as a student of UFSC, to deposit this document in the said Institutional Repository, for being it a Final Program Dissertation (*"Trabalho de Conclusão de Curso"*), in accordance with the *Resolução Normativa n° 126/2019/CUn*.

Tommy Venek

Tommy Venek
gemineers GmbH

ABSTRACT

This document describes the author's internship at the company gemineers GmbH in Aachen, Germany, during 2023, focusing on enhancing the company's machining quality assessment platform through the integration of an advanced cutting tool monitoring system, developed in response to the high precision and efficiency requirements across aerospace, automotive, and electronics industries. The developed system's infrastructure allows users to efficiently manage cutting tools, featuring a 3D viewer for real-time tool representation, enhancing user experience and safeguarding against errors in wear estimation. The integrated lifetime monitoring solutions aim to optimize tool usage, with one approach analyzing tool usage at each surface point, effectively extending tool lifespan by over 80%, and the other seeking to identify the tool's wear stage through acquired and generated data. This data-driven approach establishes correlations between wear and process data, offering valuable insights for a proactive and precise tool replacement strategy.

Keywords: Machining. Digital Twin. Tool Wear.

RESUMO

Este trabalho documenta o estágio do autor na gemineers GmbH em Aachen, Alemanha, durante 2023, com foco em aprimorar a plataforma de avaliação de qualidade de usinagem da empresa por meio da integração de um sistema avançado de monitoramento de ferramentas, idealizado em resposta às exigências rigorosas de precisão e eficiência nas indústrias aeroespacial, automotiva e eletrônica. A infraestrutura desenvolvida do sistema permite que os usuários gerenciem eficientemente ferramentas de corte, apresentando um visualizador 3D para representação em tempo real das ferramentas, aprimorando a experiência do usuário e protegendo contra erros na estimativa de desgaste. As soluções integradas de monitoramento de vida útil visam otimizar o uso de ferramentas, com uma abordagem analisando o uso da ferramenta em cada ponto de superfície, estendendo efetivamente a vida útil em mais de 80%, e a outra buscando identificar o estágio de desgaste da ferramenta por meio de dados adquiridos e gerados. Essa abordagem orientada por dados estabelece correlações entre desgaste e dados de processo, oferecendo insights valiosos para uma estratégia proativa e precisa de substituição de ferramentas.

Palavras-chave: Usinagem. Gêmeo Digital. Desgaste de Ferramentas.

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
CAD	Computer Aided Design
CI/CD	Continuous Integration and Continuous Deployment
CNC	Computerized Numerical Control
DXF	Drawing Exchange Format
HTTP	Hypertext Transfer Protocol
IPT	Institute for Production Technology
MVP	Minimum Viable Product
REST	Representational State Transfer
UI	User Interface
UML	Unified Modeling Language
UX	User Experience

CONTENTS

1 INTRODUCTION

This document presents the activities conducted by the student during an internship at gemineers GmbH in Aachen, Germany, in 2023. The company specializes in providing a quality assessment platform for machining processes using digital twins.

This chapter serves as an introduction to the project's background, explaining the motivation behind it, its objectives, and providing an overview of the document's structure.

1.1 MOTIVATION

In today's competitive manufacturing scenario, precision, efficiency, and cost-effectiveness are crucial for gaining a competitive edge. Industries like aerospace, automotive, and electronics, which heavily rely on machining processes, continuously seek innovative methods to improve their production capabilities.

The demand for precision in machining processes is particularly strict, as even minor deviations can have profound impacts on the quality of the final product. Achieving and maintaining such precision is a complex task, requiring a delicate interplay between cutting-edge technology and skilled craftsmanship.

One critical factor that significantly influences the success of these machining processes is the performance of cutting tools. They play a vital role in shaping raw materials into intricate components, and their condition directly impacts the quality of the end product and the overall productivity of the manufacturing process.

Presently, cutting tool costs rank among the highest expenses in machining processes, occasionally exceeding 35% of unit production costs [1]. Consequently, this document focuses on a project aimed at quality monitoring of cutting tools and wear prediction, specifically to prevent premature tool replacement or using tools beyond their intended lifetime.

1.2 OBJECTIVES

1.2.1 General Objective

The main objective of the project described in this document is to implement a system for monitoring cutting tools into the gemineers software. This system will encompass fundamental functionalities, including the creation, editing, and deletion of cutting tools, alongside advanced features such as usage monitoring and tool wear estimation. By focusing on these actions, the system aims to deliver a more refined tool replacement strategy, ultimately leading to the optimization of production processes, cost reduction, and the assurance of peak performance and reliability for clients.

1.2.2 Specific Objectives

To achieve its goal, the system will be structured into multiple user-friendly graphical interfaces, enabling users to efficiently interact with and monitor their cutting tools. These interfaces will be connected with the rest of the software, particularly the database, ensuring data flow and accessibility.

Furthermore, an innovative tool wear modeling methodology will be developed and integrated into the software. This methodology represents a crucial component, enabling the system to accurately estimate tool wear and subsequently enhance decision-making related to maintenance and replacement.

1.3 STRUCTURE OF THIS DOCUMENT

- **Chapter 1 - Introduction**
  This chapter provides an overview of the project's context, outlines the objectives, and explains the motivation behind the project.
- **Chapter 2 - The Company**
  In this chapter, an exploration of the company's context is provided.
- **Chapter 3 - Theoretical Background**
  The chapter explores the technologies used in the project's development and provides insight into the current state of the subject matter.
- **Chapter 4 - Proposed Methodology**
  This chapter explains the different approaches for project planning and execution.
- **Chapter 5 - Development**
  A detailed description of all the steps involved in the project's development is presented in this chapter.
- **Chapter 6 - Conclusion**
  This chapter summarizes the project's achievements and offers recommendations for future enhancements.

2 THE COMPANY

gemineers GmbH is a spin-off startup from the Fraunhofer IPT, founded in July 2021. The company is based in Aachen, Germany, and its primary objective is to accelerate the digitalization of machining processes.

The gemineers product is a software that enables clients to monitor machining processes using digital twin technology. By collecting data directly from the machine tools, the software offers complete digitalization of machining products and processes. This facilitates easier quality assessment and product evaluation, in addition to providing a digital record of all machining processes.

Figure 1 – gemineers GmbH logo.

Source: [2]

2.1 FRAUNHOFER IPT

The Fraunhofer IPT, depicted in Figure 2, is one of the institutions within the Fraunhofer Society, a German research organization with over 70 institutes distributed across Germany, each specializing in various fields of applied science. Some of the departments at Fraunhofer IPT and their areas of expertise include:

- High-Performance Cutting
  This department is where gemineers GmbH originated. Its areas of focus encompass artificial intelligence and digital twin technology, both applied to the field of cutting [3].

- Fine Machining and Optics
  This department conducts research and development in technologies for precision component production. Some of its technology portfolio includes automated fine machining, diamond cutting, ultra-precision grinding, polishing, nano-structuring, and molding of high-precision glass optics [4].

- Fiber-Reinforced Plastics and Laser System Technology
  The primary competence of this department lies in developing special-purpose

machines for high-performance materials and laser systems, which, for instance, enable the optimization of conventional production systems and machines for applications such as sheet metal working [5].

Figure 2 – Fraunhofer IPT.

Source: [6]

3 THEORETICAL BACKGROUND

This chapter introduces the fundamental concepts necessary for comprehending the work presented in this document.

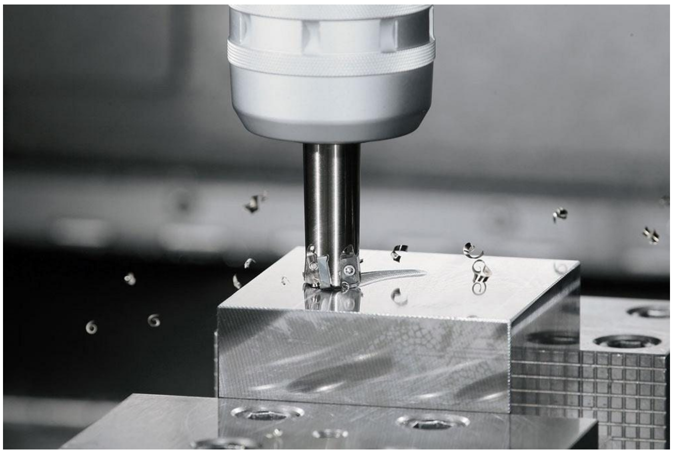
3.1 MACHINING PROCESSES

Machining is the process of shaping and sizing materials, typically metals, using specialized equipment known as machine tools, responsible for material removal. In modern manufacturing, machining is integrated with Computerized Numerical Control (CNC) technology, which involves the use of pre-programmed computer software to precisely control the movement of machine tools. There are three primary categories of machining processes: milling, drilling, and turning.

3.1.1 Milling Processes

Milling is a machining process widely used for shaping and finishing materials. It involves a rotating cutting tool that progressively removes material from a stationary workpiece and can achieve various surface finishes and complex shapes [7]. An example of a milling process is illustrated on Figure 3.

Figure 3 – Milling tool in action, removing material from a workpiece.



Source: [7]

3.1.2 Drilling Processes

Drilling is a machining process employed to create cylindrical holes in solid materials. This technique employs a rotating cutting tool, commonly referred to as a drill bit, that applies axial force to penetrate the workpiece [7]. Figure 4 depicts a drilling process.

Figure 4 – Drilling process creating holes in a workpiece.

Source: [7]

3.1.3 Turning Processes

Turning processes involve a stationary cutting tool while the workpiece rotates, being commonly used for cylindrical shaping and creating symmetrical components. The workpiece is secured on a spindle and rotated, allowing the cutting tool to remove material and shape the final product. One example of a turning process can be seen on Figure 5.

Figure 5 – Turning process in action.

Source: [7]

3.2 MACHINING TOOLS

Since the project described in this document is mainly focused on the monitoring of cutting tools, it is essential to understand their different types and applications.

3.2.1 Milling tools

- Bull nose end mill
  This type of tool is designed for general milling operations and features a rounded tip, making it well-suited for milling tasks that involve rounded corners [8].
- Ball end mill
  Tools of this type feature a spherical tip and are rotated against the workpiece to generate round-bottomed slots, pockets, or to machine intricate shapes without sharp corners [9].
- Chamfer mill
  Chamfer mills are employed to eliminate sharp edges that frequently result from other machining operations on workpieces. This creates sloped surfaces, known as chamfers, enhancing both structural integrity and visual appeal for the end user [10].
- Taper mill
  This type of tool is employed for several tasks, including creating angled walls and reaming holes to give them a conical shape [11].

3.2.2 Drilling tool

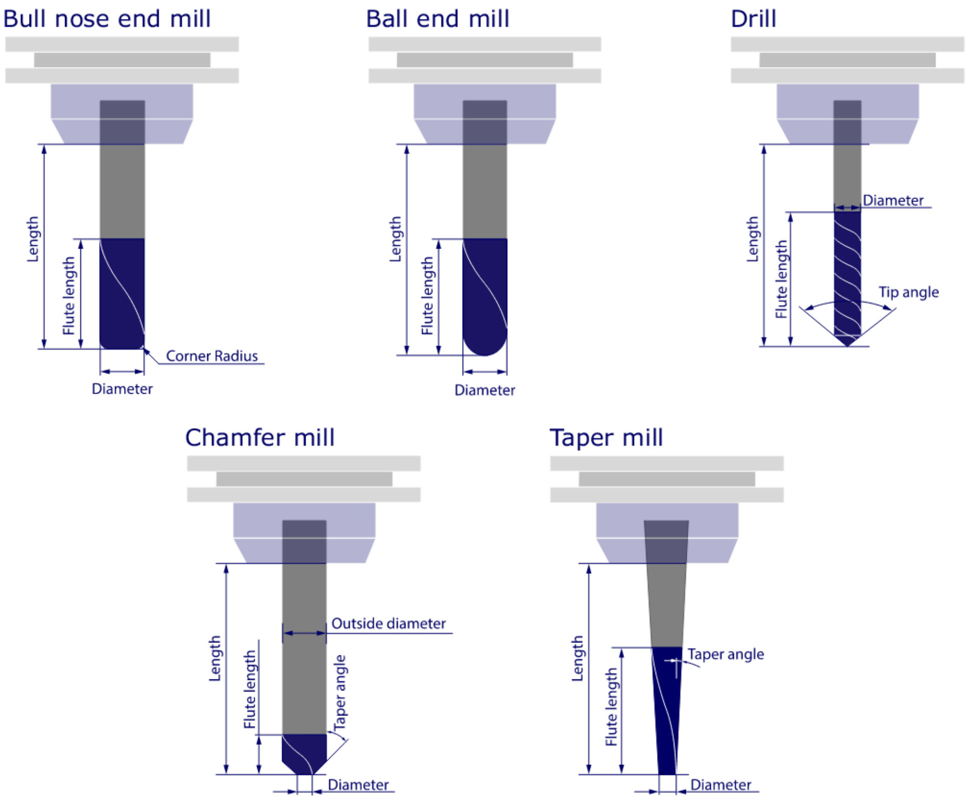
Unlike milling tools, the gemineers software does not differentiate between various types of drilling tools, following a standard model characterized by the specific parameters depicted in Figure 6, which also showcases the features regarding milling tools.

3.3 TOOL WEAR

In machining processes, cutting tools endure substantial forces from the friction between the tool and workpiece, resulting in the overheating of both the tool and workpiece in the deformation and friction zones. Given the chemically active nature of the contact surfaces, the cutting process gets involved in complex physical-chemical processes that lead to tool wear, evident in the progressive erosion of particles from the tool surface [12].

Tool wear in machining processes can lead to various consequences, including increased cutting forces, elevated temperatures, poor surface finish, and even tool

Figure 6 – Cutting tools and their key attributes.



Source: gemineers' archive.

breakage. It is classified into mechanical wear, thermochemical wear, chemical wear, and galvanic wear, with additional classification based on patterns and locations. Common types include crater wear, that results from intensive friction at the chip-tool interface, and flank wear, attributed to high cutting speed and rubbing of flank faces on work surface [13].

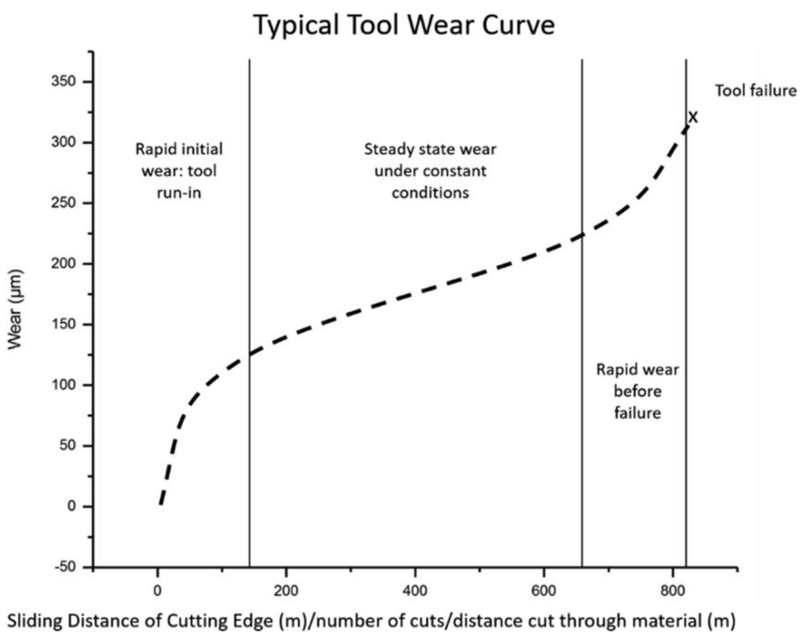
The tool wear behavior, especially flank wear, is commonly described over usage as having three distinct regions of growth: the initial rapid wear, occurring when the tool's cutting edge is blunted from a very sharp point; the steady-state wear, involving a gradual and stable increase in wear over time; and the accelerated wear stage, manifesting when the tool becomes severely worn and approaches failure [14]–[18]. The usage is typically measured by cutting time or the length traveled by the tool. This curve, referred to as the Tool Wear Typical Curve, is depicted in Figure 7.

3.3.1 Tool Life

The estimated lifetime of a cutting tool is often provided by the manufacturer or supplier, usually being included in the product documentation. This expected tool life frequently is based on various factors such as the type of material being cut, cutting speed, feed rate, and other machining conditions.

This estimated tool life serves as a practical reference for determining when a tool should be replaced, especially considering that machine tools usually measure the duration of machining operations.

Figure 7 – Typical Tool Wear Curve.



Source: [14]

3.4 DIGITAL TWIN TECHNOLOGY

A digital twin is a virtual representation of a physical process or object. These digital duplicates mimic real-world processes and physical objects, offering several opportunities across a wide range of applications. They are designed to replicate the behavior of their physical counterparts, enabling the prediction of product performance, quality optimization, and a deeper understanding of complex systems.

Digital twins essentially comprise a real-time data acquisition system and the application of mathematical models to emulate the actual process or object [19]. This technology extends beyond mere simulation, allowing for real-time analysis of intricate systems and providing valuable insights that improve efficiency and decision-making [20].

Figure 8 – Digital twin of a blisk in the gemineers software.

Source: gemineers' archive.

3.5 SOFTWARE DEVELOPMENT

3.5.1 Front end

Front-end development focuses on designing and coding visual user interfaces and their connections to data and services. In this project, the selected framework was Next.js, built upon the React library, because of its high performance, coupled with its widespread popularity, active community, and extensive documentation.

React applications are written in JavaScript and are composed of components, self-contained interface elements with their own logic and appearance. Components can range from small elements like buttons to entire pages, and are typically developed as separate, reusable, and scalable units.

Incorporated into the project were several other third-party JavaScript libraries, which include:

- Material UI
  This library that offers a collection of pre-styled React components with additional features. It was used to construct the foundational structure of the software's user interfaces.

- ECharts
  This library provides a wide range of interactive and highly customizable charts and graphs. It was employed in several sections of the software to provide data visualization.

- Three.js
  Three.js is a library dedicated to the development of 3D visualizations, offering re-

sources like geometries, meshes, and animations. In the gemineers software, it primarily serves to render toolpaths of processes, parts, and digital twins. The library plays a significant role in presenting visualizations of the user's tools throughout this project.

3.5.2 REST API

An API is a set of rules and conventions that define how different software applications or devices can establish connections and exchange information. Among the various types of APIs, the REST API follows a set of key principles that facilitate efficient communication between clients and servers [21].

The fundamental principles of REST include:

- Stateless Communication
  REST APIs treat each request as an independent and self-contained message without relying on previous interactions.
- Uniform Interface
  REST enforces a consistent and uniform set of interactions for all resources, promoting simplicity and predictability.
- Client-Server Decoupling
  REST separates the client and server components, allowing them to evolve independently and improving scalability and flexibility.
- Layered System
  REST systems can be organized into multiple layers, such as load balancers and caching systems, for added flexibility and improved performance.

In the gemineers software, the REST API was developed using a JavaScript-based framework. It was chosen because of its powerful capabilities and also because it uses the same programming language as the software's front end, which makes it easier for developers to work on both services.

3.5.2.1 HTTP

Hypertext Transfer Protocol (HTTP) is the fundamental protocol for data communication on the World Wide Web. It facilitates the exchange of information between clients (user devices) and servers (remote machines), operating on a request-response model [22].

In the gemineers software, all communications between services and the database, such as reading and writing, are executed through HTTP requests from the services to the REST API. Specific methods define these client-server interactions, including:

- GET: Retrieves data from a specified resource. It is used, for example, by the software's front end to read and view information from the database.
- POST: Submits data to be processed to a specified resource. It is commonly used, for example, by the software's front end to register new cutting tools into the database.
- PUT: Updates a specified resource or creates one if it does not exist.
- DELETE: Removes a specified resource.

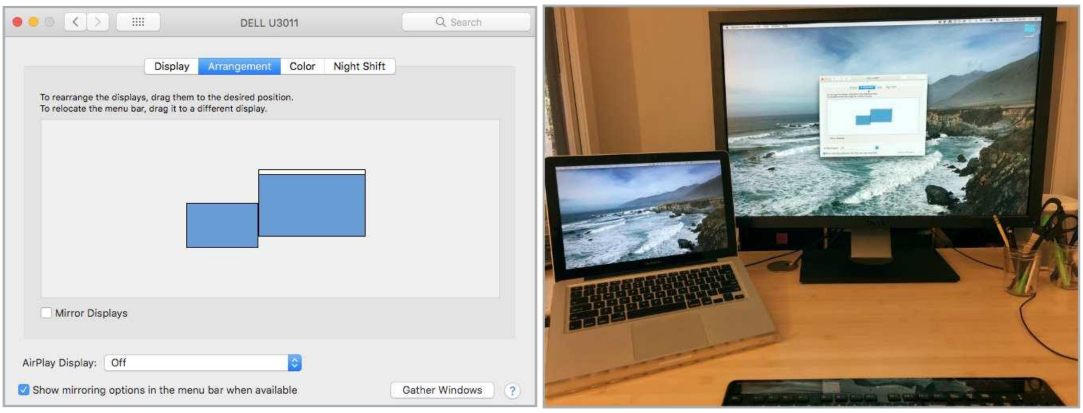
3.6 UX/UI DESIGN

User Interface (UI) Design involves the visual aspects of interfaces, such as color schemes, typography, and element placement. In contrast, User Experience (UX) Design considers the overall user experience, combining UI elements to ensure that interactions are both visually appealing and user-friendly, and optimizing the interface for efficiency. Together, UI and UX Design aim to build digital experiences that are both visually pleasing and functionally intuitive.

There are several principles and guidelines that help in the design of intuitive and efficient interfaces, such as:

- Responsiveness
  Interfaces should be responsive, adapting to different screen sizes and resolutions [23].
- Match Between the System and the Real World
  Systems should use familiar language and concepts, aligning with users' real-world expectations [24].
- Natural Mapping
  Controls should correspond to the desired outcome, making systems easier to learn and remember [25]. One example of this principle is depicted in Figure 9.
- Consistency and Standards
  Interfaces should follow established conventions and maintain consistent patterns [26].
- Error Prevention
  Good interfaces proactively prevent user errors, while also providing clear error messages and documentation when necessary [27].

Figure 9 – Example of the Natural Mapping principle.



Source: **[25]**

3.7 PYTHON

Python is an object-oriented programming language known for its simple syntax and extensive standard libraries, making it well-suited for tasks such as testing, prototyping, and data analysis. Although this language was not directly used in the software development of this project, several Python modules were utilized to perform data analysis and create visualizations. Some of those libraries include:

- pandas
  pandas is a data manipulation and analysis module. It provides data structures optimized for organizing and analyzing data and simplifies tasks such as data cleaning, exploration, and transformation.

- Plotly
  This module enables the creation of interactive charts. It supports several chart types, making it suitable for a wide range of data visualization needs.

- NumPy
  NumPy is a library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

These modules were chosen for their robust capabilities and extensive documentation.

4 METHODOLOGY

This chapter describes the methods and steps followed during the planning and development of this project.

4.1 SCRUM FRAMEWORK

Scrum is an agile project management framework designed to enhance collaboration, adaptability, and continuous improvement. It promotes iterative development through fixed-length cycles known as sprints. In this project, the decision was made to adopt a two-week sprint cycle.

4.1.1 Sprint Planning

Sprint Planning marks the beginning of each sprint and involves the entire development team. During this event, the team collaboratively selects backlog items based on priority and defines the tasks to be accomplished during the sprint.

4.1.2 Daily Scrum

Daily Scrums are brief, daily meetings where team members provide updates on their progress, discuss any challenges or impediments, and coordinate efforts. These meetings ensure that everyone is informed, aligned, and can quickly address any emerging issues.

4.1.3 Sprint Review

This event is conducted to showcase the completed work to stakeholders. It creates an opportunity for them to provide feedback and ensures that the delivered increment aligns with the project goals.

4.1.4 Sprint Retrospective

Following the Sprint Review, the team engages in a Sprint Retrospective. This reflective session allows the team to discuss what went well, identify areas for improvement, and plan adjustments for the next sprint.

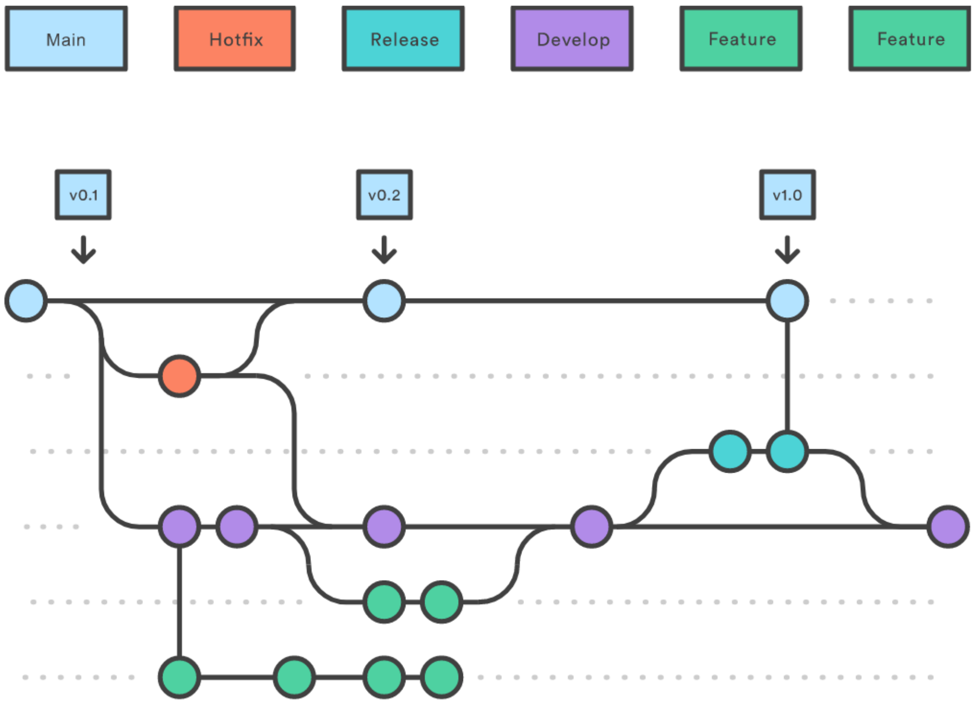
4.2 GIT AND VERSION CONTROL

Git, a distributed version control system, has played an important role in the software development of this project. Applied to both front-end and REST API development, Git enables multiple developers to work on a project simultaneously, allowing for efficient collaboration, tracking of changes, and maintaining a complete history of the codebase [28].

In a Git workflow, a development branch refers to a separate copy of the project that each developer works on locally. This branch allows developers to make changes to files and commit those changes independently. It serves as an isolated space where features or bug fixes can be developed without directly affecting the main codebase.

The main branch, often named "Main", is the primary branch that represents the stable and production-ready version of the codebase. Once a task is complete in a development branch, developers create a merge request to propose their changes for integration into the main codebase. Figure 10 illustrates a common branch workflow using Git.

Figure 10 – Git branch workflow.



Source: [29]

Before that integration is complete, other team members review the proposed changes, providing feedback and ensuring code quality. This process helps catch potential issues early, maintains a high standard of code, and facilitates knowledge sharing among team members.

In addition to the code review, the proposed changes from a merge request also pass through a Continuous Integration and Continuous Deployment (CI/CD) pipeline, where they undergo automated testing, code analysis, and, if successful, deployment to a staging environment. This pipeline ensures that the changes are compatible with the existing codebase and do not introduce new issues.

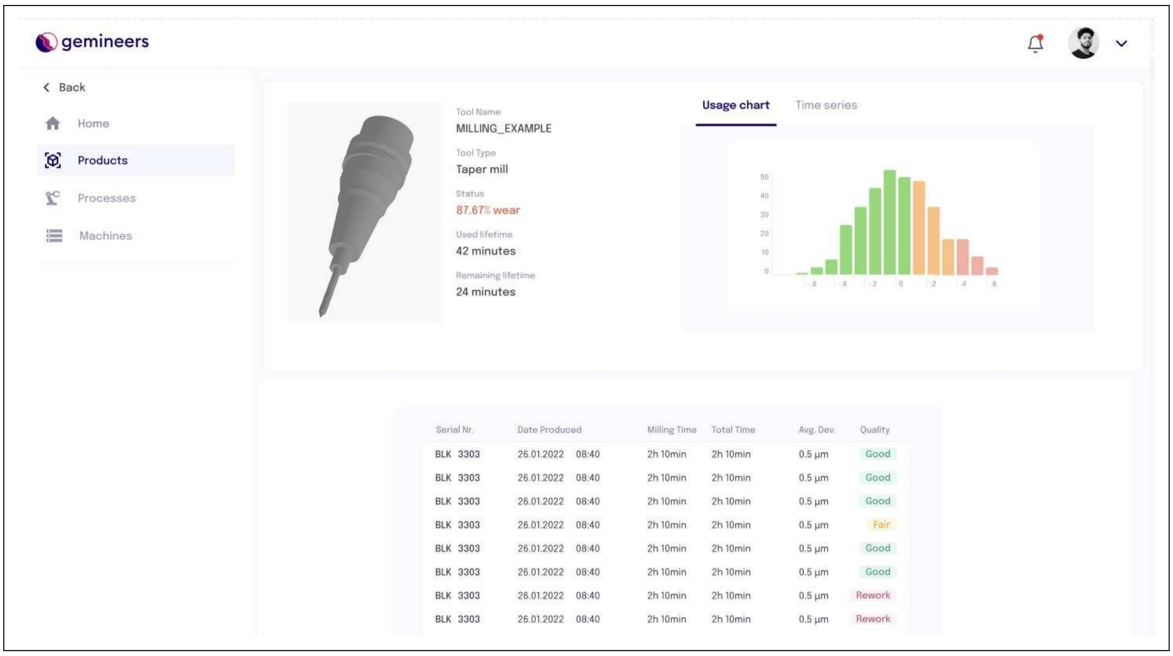
This set of processes and steps promotes a structured and organized approach to managing code changes, contributing to a more robust and maintainable software development workflow.

4.3 INTERFACE PROTOTYPING

For the front-end development phase, the process began with the utilization of Figma, a collaborative design tool, to create interface prototypes. After generating these prototypes, external feedback was sought to validate the design. Figure 11 shows one of the interfaces prototypes.

Following the validation stage, the focus shifted to constructing a Minimum Viable Product (MVP) of the interface. The development process involved a continuous iteration loop based on received feedback, refining and enhancing the interface until reaching the final version. This approach ensured that the interface met the necessary requirements and user expectations, contributing to the overall success of the front-end development.

Figure 11 – Prototype of one of the software's interfaces.



Source: Author's archive (2023).

4.4 UNIT TESTS DEVELOPMENT

A crucial step in REST API development was creating unit tests, for systematically verifying code sections for intended functionality. These tests, covering diverse scenarios, ensured the reliability of API methods and endpoints.

This test-driven strategy not only detected bugs early in development but also contributed to building a more robust and maintainable REST API. Developers could manually run these unit tests, and they were automatically integrated into the CI/CD pipeline, ensuring software stability.

4.5 SOFTWARE ARCHITECTURE

4.5.1 UML

UML is a standardized modeling language in the field of software engineering. It provides a visual representation of a system's architecture, design, and implementation using a set of diagrams [30].

Through its visual representation, this language facilitates communication among stakeholders and provides a clearer understanding of a system, besides acting as documentation for software projects.

During this project, the author utilized UML, particularly employing Class Diagrams (depicting database structure and relationships) and Activity Diagrams (illustrating the workflow of specific functions). Examples of these diagrams can be found in Appendix B.

4.5.2 SOLID

SOLID is a set of five design principles, followed during the front-end development of the project, that aim to guide developers in creating maintainable, scalable, and flexible software [31]. Although SOLID principles were formulated in the context of object-oriented programming, their fundamental concepts can be applied to various programming paradigms, including event-driven and structured programming, the adopted paradigms in the project's front end.

The application of SOLID principles within the React context is explained below:

- Single Responsibility Principle
  In the React context, this extends to components focusing on rendering UI or handling specific aspects of the application's logic, instead of having multiple tasks [32].

- Open/Closed Principle
  According to this principle, a class or module should be open for extension but

closed for modification. In React development, this is achieved through the clever use of props (React's mechanism for passing data and configuration between components) to extend functionality without directly modifying existing code [32].

- Liskov Substitution Principle
  In the React context, this principle asserts that components extending or based on other components should share similar behavior and implementation and be interchangeable without introducing errors. It promotes a consistent and compatible structure, enhancing the maintainability and extensibility of the React application [32].

- Interface Segregation Principle
  In the context of React development, this principle suggests that components should not have access to unnecessary information. Instead, each component should receive only the specific props it requires, promoting a more modular and focused design [32].

- Dependency Inversion Principle
  In the React context, this principle encourages designing components that are independent from each other by promoting the use of abstractions and avoiding direct dependencies, allowing a modular and loosely coupled architecture [32].

In conclusion, the combination of Scrum and UML facilitated planning of tasks, while following software practices and principles like SOLID ensured a robust and scalable project development. Moreover, feedback-oriented development strategies, such as code reviews and interface prototyping, helped align the project with expectations, and the integration of error-preventing measures, such as version control and unit tests, improved the project's reliability and security.

5 DEVELOPMENT

5.1 DATABASE CONFIGURATION AND REST API DEVELOPMENT

In order to enable users to register tools within the system, the REST API needed to be configured to receive cutting tool data from the software's front end and insert it into the database, in addition to providing the capability to read, delete, and update tool data within the same database.

5.1.1 Cutting Tool Model

The creation of the cutting tool model took into consideration the following attributes:

- Tool name
- Tool type (milling, drilling, or turning)
- Tool subtype (e.g., ball end mill, chamfer mill, taper mill, etc.)
- Length
- Flute length
- Inside length (length of the tool section inside the holder)
- Diameter
- Helix angle
- Number of teeth
- Shank material
- Tool holder sections (explained in 5.2.2.1)
- Corner radius (for bull nose end mill and taper mill)
- Taper angle (for taper mill and chamfer mill)
- Tip angle (for drilling tools)
- Lower diameter (for chamfer mill)

With these characteristics, the software is capable of registering all types of tools that the company intends to use.

5.1.2 Tool Usage Model

To facilitate front-end access to specific operational data, a dedicated tool usage model was created. It includes the following data:

- Tool name
- Operation ID
- Tool profile X and Y coordinates
- Total contact duration per tool surface point
- Total volume removed per tool surface point
- Total length traveled per tool surface point
- Temporal signals, such as spindle load variations throughout the operation

5.2 INTERFACES

This section outlines the development steps for the user-friendly interfaces related to cutting tool management.

5.2.1 Tools Main Page

This page serves as a platform for displaying all tools registered by the currently authenticated user, presented in either a grid or list format, as shown in Figure 12. Users can interact with each tool, allowing them to modify tool parameters, monitor usage, or create and delete tools as needed.

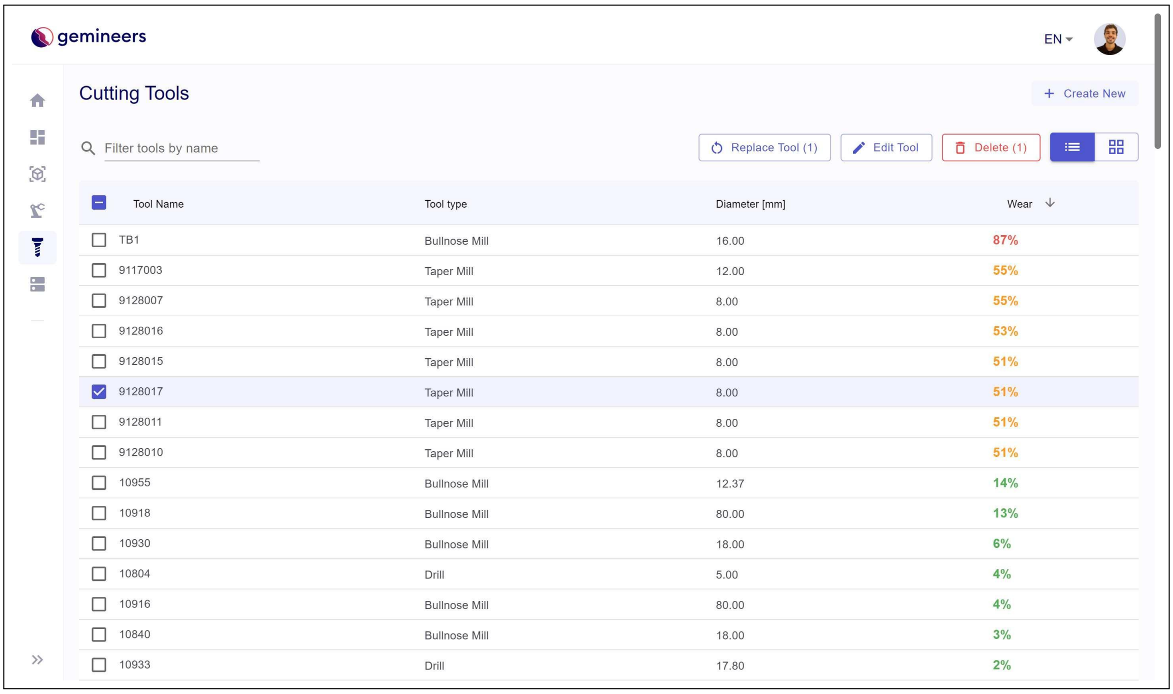
In the development of this page, it was necessary for the software's front end to fetch the list of tools from the REST API and subsequently render a list containing several information regarding each tool.

5.2.1.1 Tool creation and edition

Initially, the approach to enable users to create new tools and edit those already registered was to design a modal (an element that displays in front of and deactivates other page content [33]) containing the tool fields, as illustrated in Figure A.1, in Appendix A.

The creation and editing of tools were accomplished through POST and PUT HTTP requests originating from the software's front end to the REST API, following the tool model schema described in Section 5.1.1. The use of modal windows for this purpose was meant to be temporary, as the plan was to implement a dedicated page for these tasks, as detailed in Section 5.2.2.

Figure 12 – Tools Main Page in list layout.



Source: Author's archive (2023).

5.2.2 Tool Definition Page

This page, replacing the modal window logic in the Tools Main Page, allows users to create and edit tools and can be seen on Figure 13. The interface features a 3D viewer, enabling users to view a virtual representation of the tool in real-time.

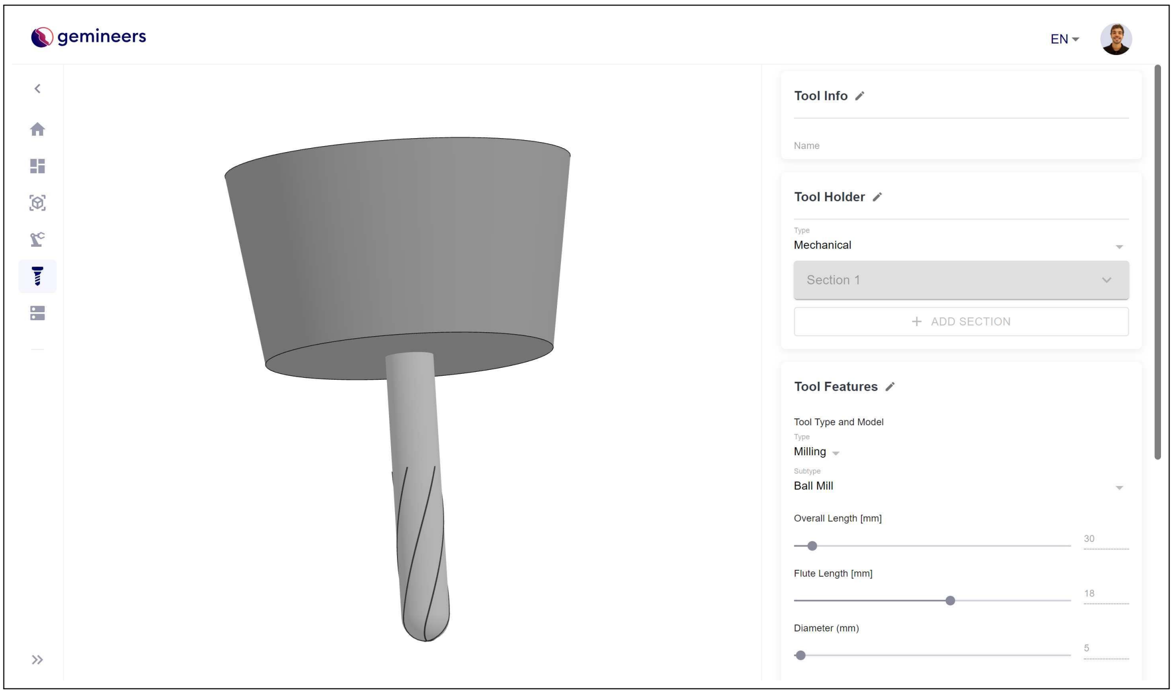
The initial version of this page allowed users to provide all the necessary data about the tool geometry, with the fields adjusting based on the selected tool type. Additionally, the user could also configure basic information about the tool holder, such as height and diameter (where the holder's shape was simplified as a truncated cone).

Several geometric shapes, such as cylinders, spheres, and cones (all provided by the Three.js library), are combined to represent the tool's geometry. The position and dimensions of each solid are calculated using basic geometry and trigonometry applied to the user-defined parameters outlined in the schema in Section 5.1.1.

5.2.2.1 Tool holder

In order to enable the users to define complex geometries for tool holders, which may assume a variety of shapes, an approach that breaks down the holder's design into various sections, such as cylinders or truncated cones, has been implemented, as illustrated in Figure 14.

Figure 13 – Tool Definition Page, containing a 3D viewer and a sidebar with fields to be filled.



Source: Author's archive (2023).

5.2.2.1.1 Deflection calculation

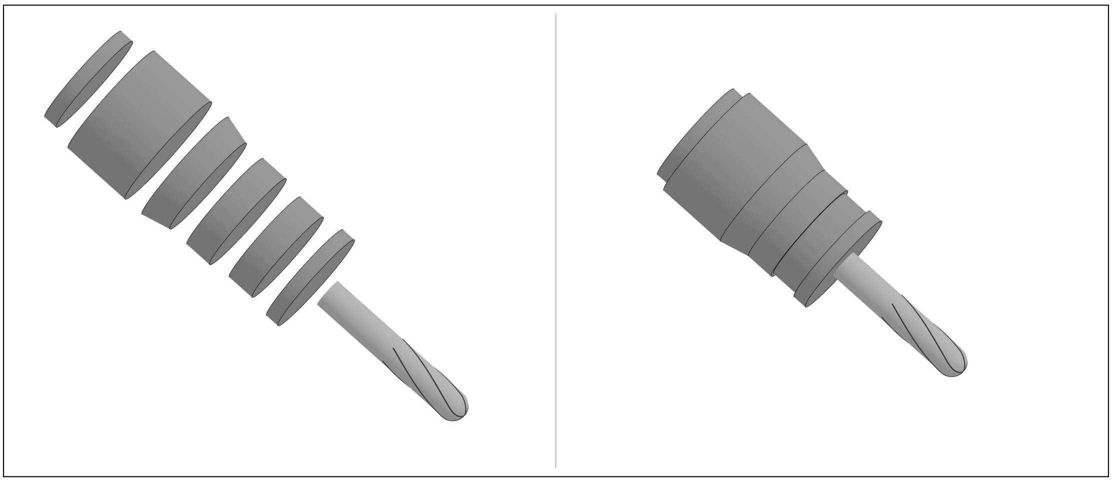
The main purpose for implementing this holder definition system is to enable accurate calculation of tool deflection, which is significantly affected by the holder's geometry. That detailed geometry and other parameters of the tool are employed to predict deflection and helps to increase the precision of the digital twin.

However, it is important to note that the deflection calculation was not conducted by the author of this paper.

5.2.2.1.2 Holder definition interface

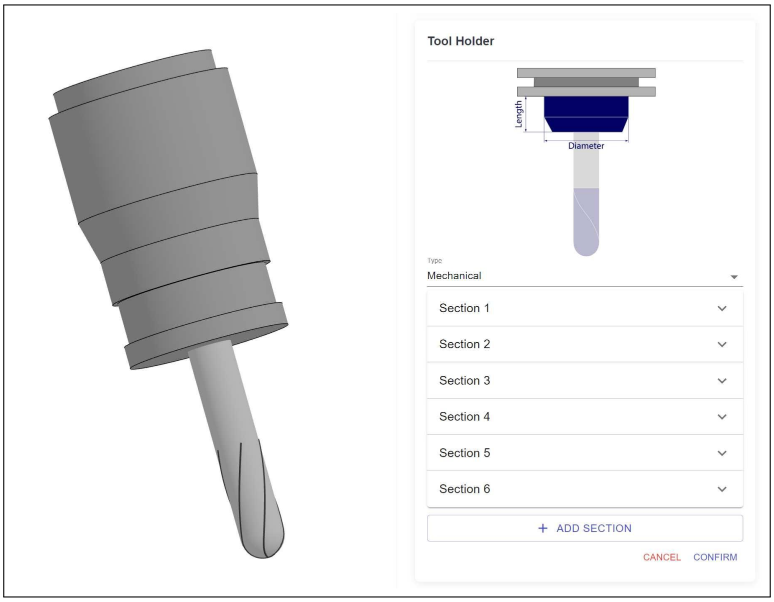
The holder definition interface employs an accordion menu, which consists of vertically stacked headers that reveal additional details when activated, with each menu item corresponding to a specific holder section, as shown in Figure 15.

Figure 14 – Design of tool holder using sections.



Source: Author's archive (2023).

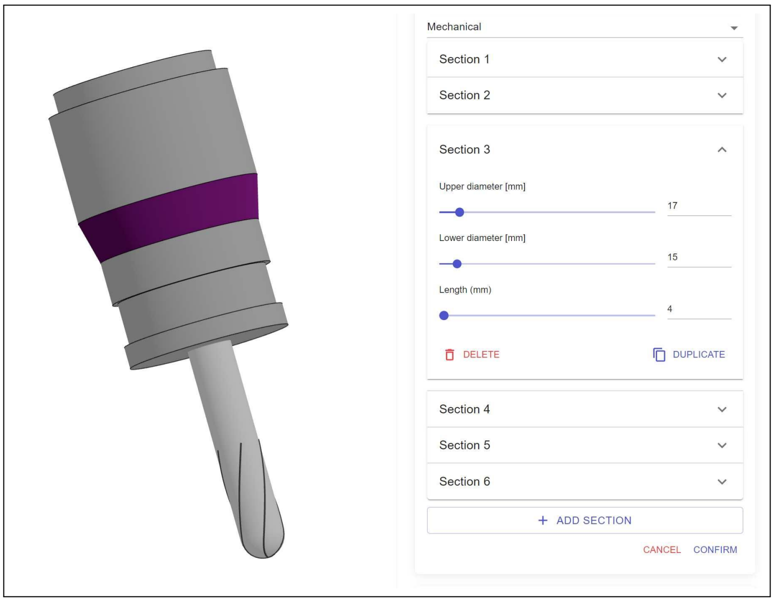
Figure 15 – Accordion menu and design of tool holder sections.



Source: Author's archive (2023).

With that structure, it is possible to set the height and the upper and lower diameters of each section, as depicted in Figure 16:

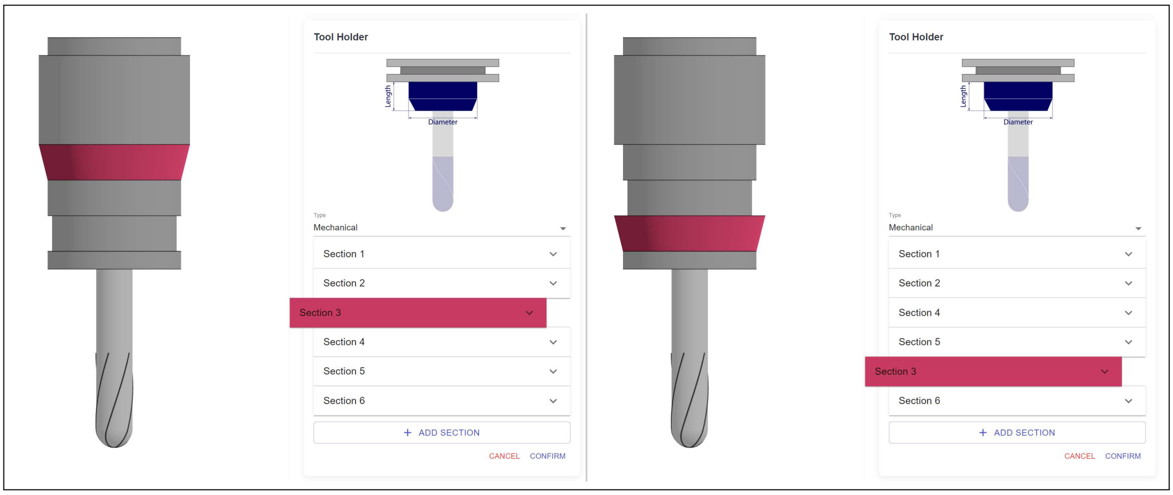
Figure 16 – Individual tool holder sections design.



Source: Author's archive (2023).

Users can also duplicate and delete sections, and rearrange them using a drag-and-drop interface. During this process, different colors assist the user in identifying the section being moved, following the Natural Mapping UX concept, as shown in Figure 17.

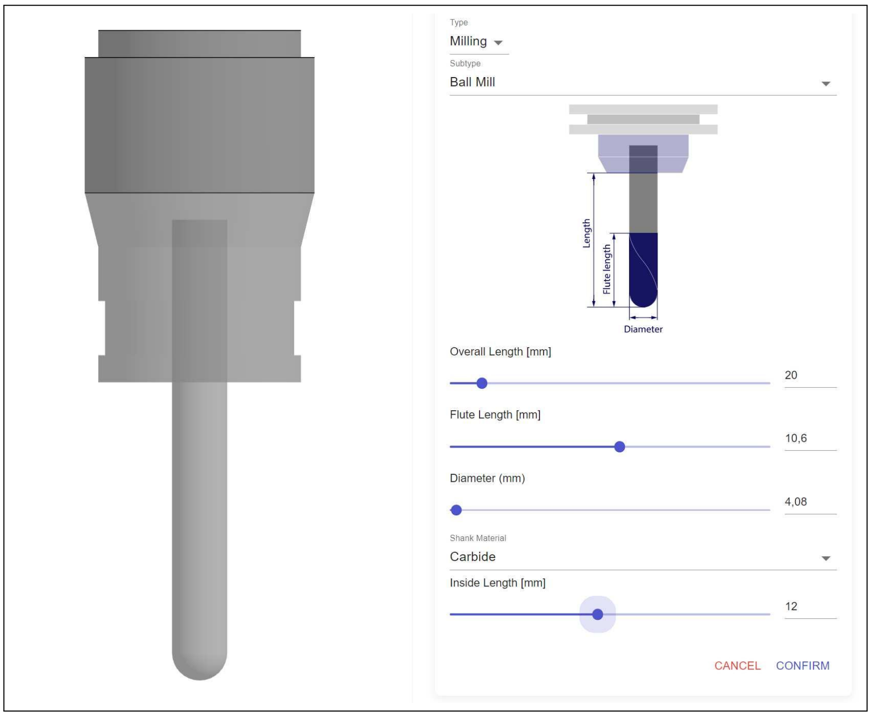
Figure 17 – Reordering of holder sections using drag-and-drop interface.



Source: Author's archive (2023).

When modifying the "inside length" feature of the tool, specific sections of the holder become partially transparent, allowing for a view of the internal portion of the tool, as depicted in Figure 18. To achieve this effect and assign different colors to sections (as shown in Figures 16 and 17), a corresponding function has been developed.

Figure 18 – Internal section of the tool.



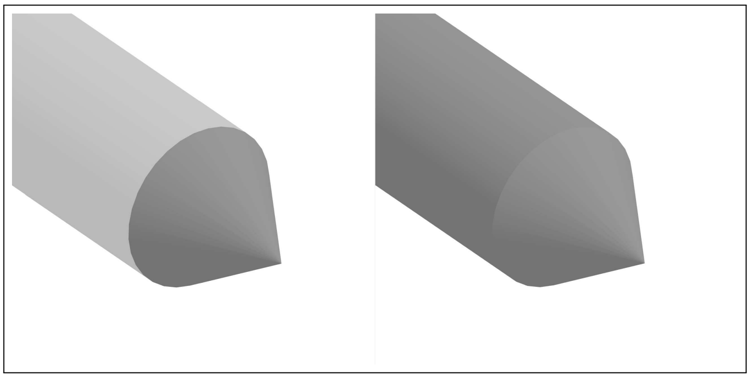
Source: Author's archive (2023).

5.2.2.2 Basic tool geometries

Drill

For the drilling tool, the geometries of a cylinder for the body of the tool and a cone for the tip of the tool were merged, as illustrated in Figure 19.

Figure 19 – Drilling tool 3D model.

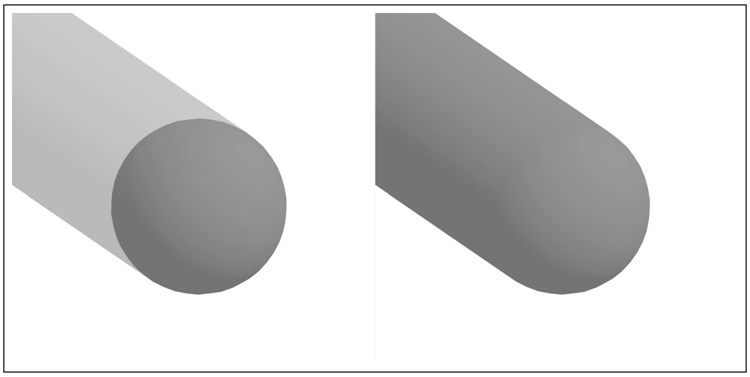


Source: Author’s archive (2023).

Ball end mill

The geometries of a sphere and a cylinder were combined to create the ball end mill, as shown in Figure 20.

Figure 20 – Ball end mill 3D model.

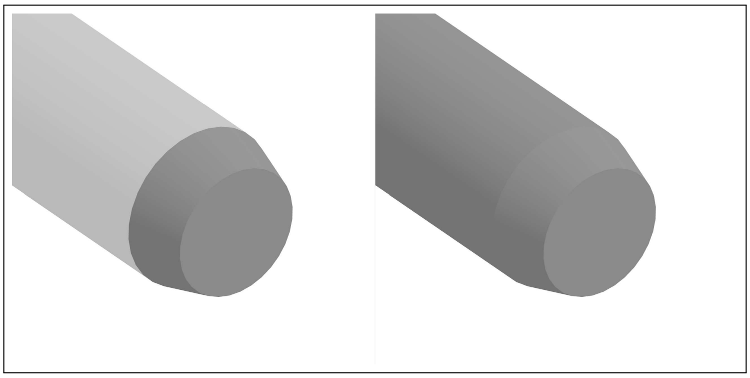


Source: Author’s archive (2023).

Chamfer mill

As illustrated in Figure 21, the chamfer milling tool is created by combining the geometries of a truncated cone and a cylinder.

Figure 21 – Chamfer mill 3D model.

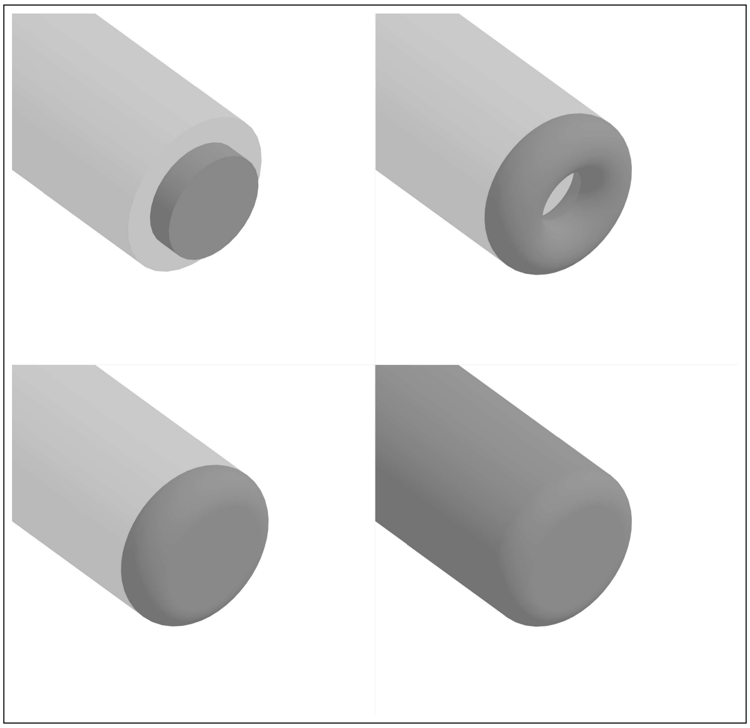


Source: Author's archive (2023).

Bull nose end mill

The bull nose milling tool combines the geometries of a torus and an auxiliary cylinder with a cylindrical body, as depicted in Figure 22.

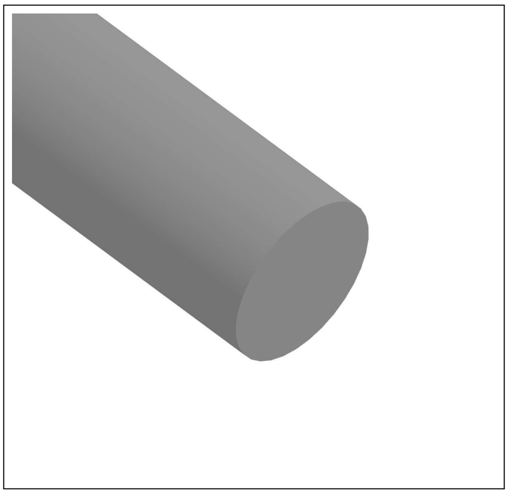
Figure 22 – Bull nose end mill 3D model.



Source: Author's archive (2023).

A bull nose end mill simplifies to a basic cylindrical geometry when the tip's corner radius is zero, as illustrated in Figure 23

Figure 23 – End mill 3D model.

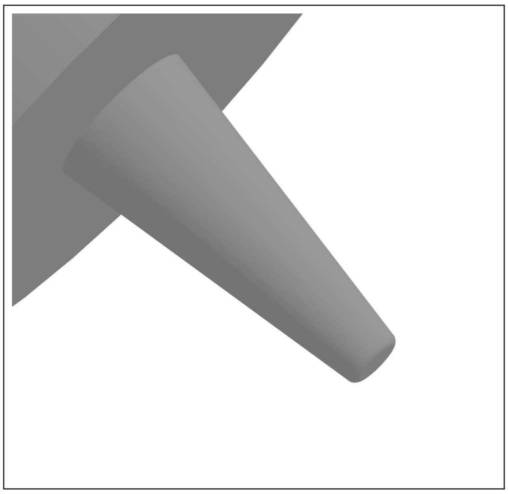


Source: Author's archive (2023).

Taper mill

In the case of the taper mill, the body of the tool is not represented by a cylinder but by a truncated cone, while the tip works exactly like the bull nose end mill, combining the torus and cylindrical geometries to produce a rounded edge, as shown in Figure 24

Figure 24 – Taper mill 3D model.

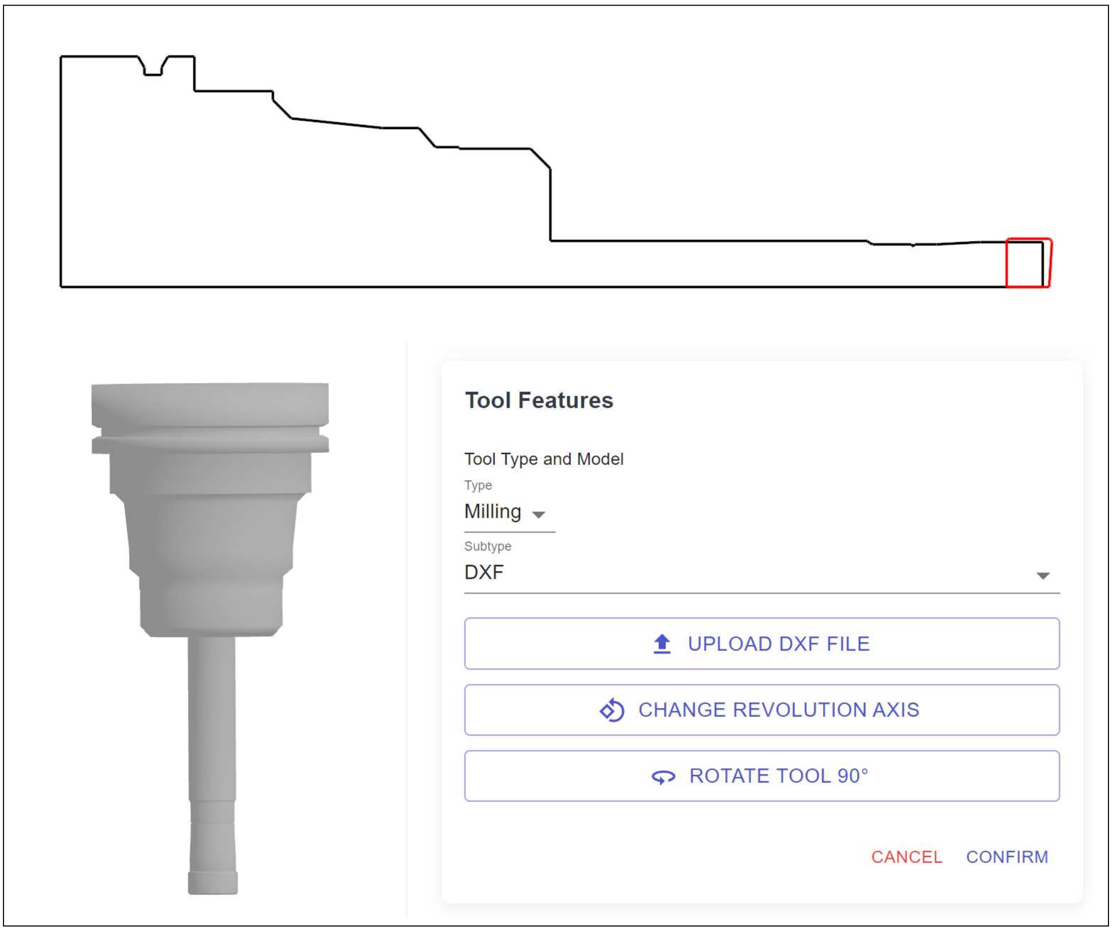


Source: Author's archive (2023).

Custom Tool

In the Tool Definition Page, users can also upload a Drawing Exchange Format (DXF) file specifying the lateral section geometry of the tool. Subsequently, the 3D viewer generates a solid of revolution to visually represent the tool, as depicted in Figure 25.

Figure 25 – Lateral section of a tool and its 3D model.



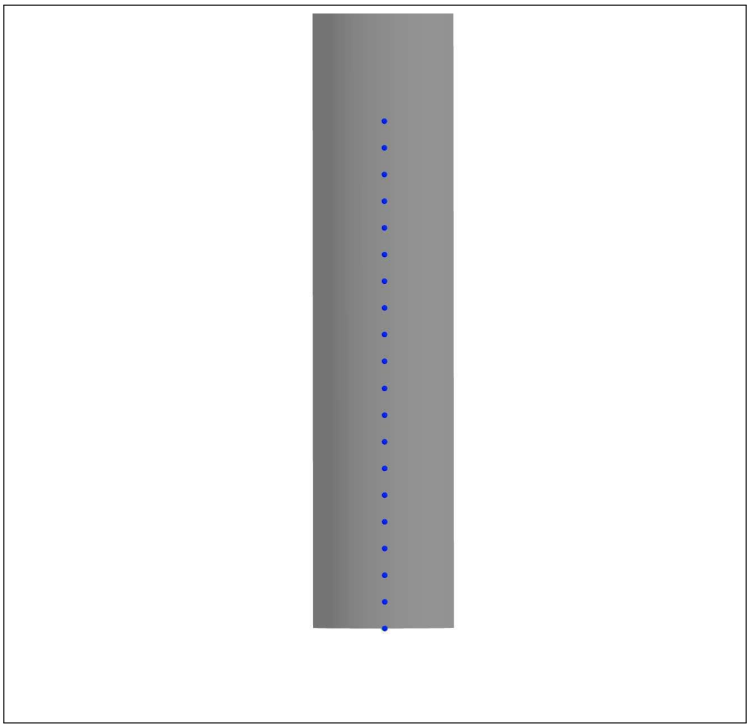
Source: Author's archive (2023).

5.2.2.3 Cutting edges

The 3D representation of the tool's cutting edges was not trivial since the Three.js library lacks built-in helix geometries. However, it does offer a tube geometry that can follow a path defined by points using an interpolation algorithm.

First, it was necessary to establish a sequence of coplanar points on the tool's surface, as shown in Figure 26.

Figure 26 – Points on the surface of the tool body.



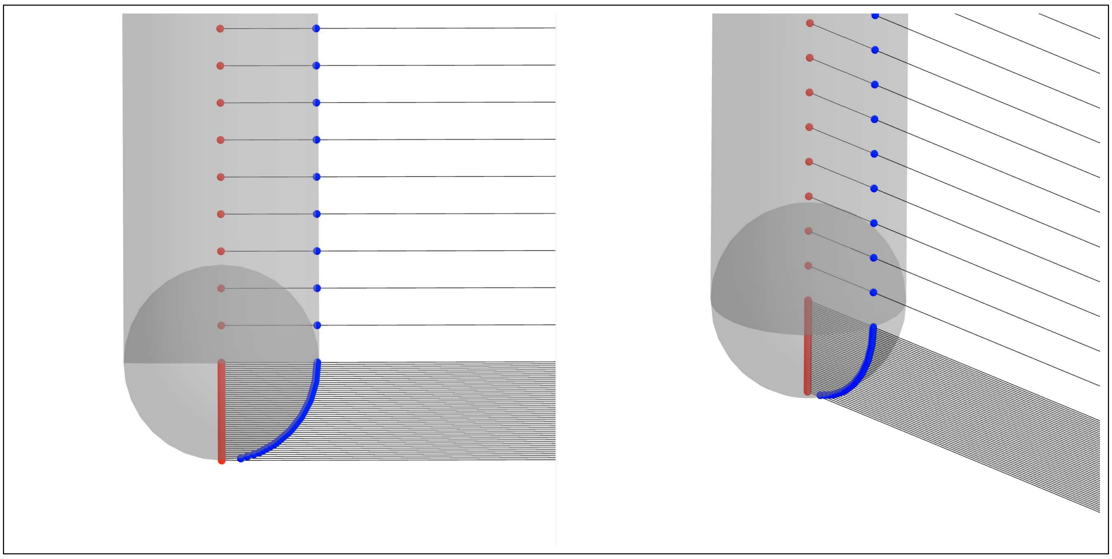
Source: Author's archive (2023).

To obtain those points, it was necessary to cast, from different points on the tool's vertical axis, several rays perpendicular to that axis. In the context of this process, ray casting, an essential technique in computer graphics and Computer Aided Design (CAD), involves tracing these rays and detect their intersections with the tool's surface. The last intersection of each ray with the tool's surface was considered to be one of the surface points.

Given the potentially complex geometry of the tool tip, a greater number of points is gathered in that region to enhance the accuracy of its representation.

This process is illustrated in Figure 27.

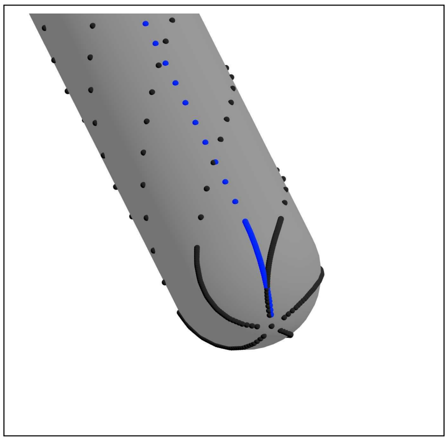
Figure 27 – Rays casted from the vertical axis of the tool define the surface points.



Source: Author's archive (2023).

Once the surface points are defined and considering an expected helix angle, it is necessary to rotate these points around the tool's vertical axis, in order to create a path for the helix to follow, as visualized in Figure 28.

Figure 28 – Rotated points representing the path of the cutting edge.

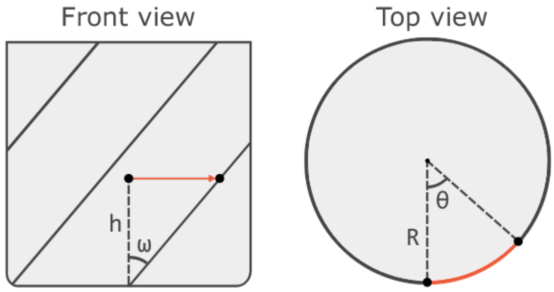


Source: Author's archive (2023).

Given that h is the height of a specific point, R is the radius of the tool on that height and ω is the desired helix angle, the rotation angle θ of that point is expressed by Equation (1), with its parameters illustrated in Figure 29.

$$\theta = \frac{h \cdot \tan \omega}{R}, \tag{1}$$

Figure 29 – Rotation of surface point to form a helix path.



Source: Author's archive (2023).

Given that x, y, and z are the coordinates before rotation, and θ is the rotation angle, the new 3D rotated coordinates x', y', and z' of each surface point are calculated using Equation (2).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

It is important to note that the Y-axis position remains constant for all points since it represents the tool's vertical axis, around which they rotate.

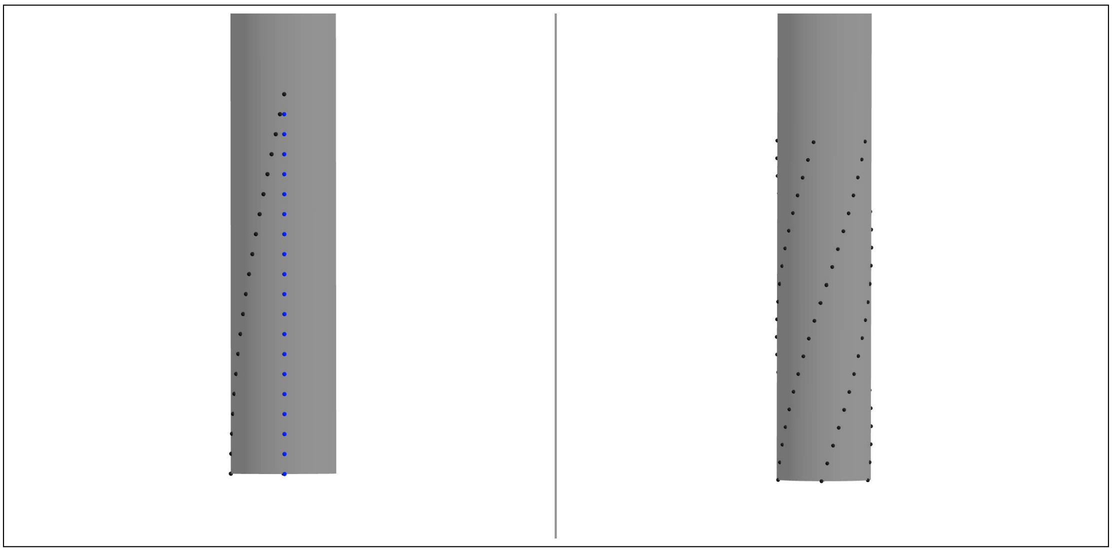
In order to have multiple teeth on the tool, an iterative offset is applied to the rotation angle θ, as shown in Equation (3). Figure 30 illustrates the rotation of surface points to form five teeth.

$$\theta = \frac{h \cdot \tan\omega}{R} + \frac{i \cdot (2 \cdot \pi)}{n}, \tag{3}$$

where:

h = height of the given point
ω = desired helix angle
R = radius of the tool on the height h
i = current iteration
n = number of teeth

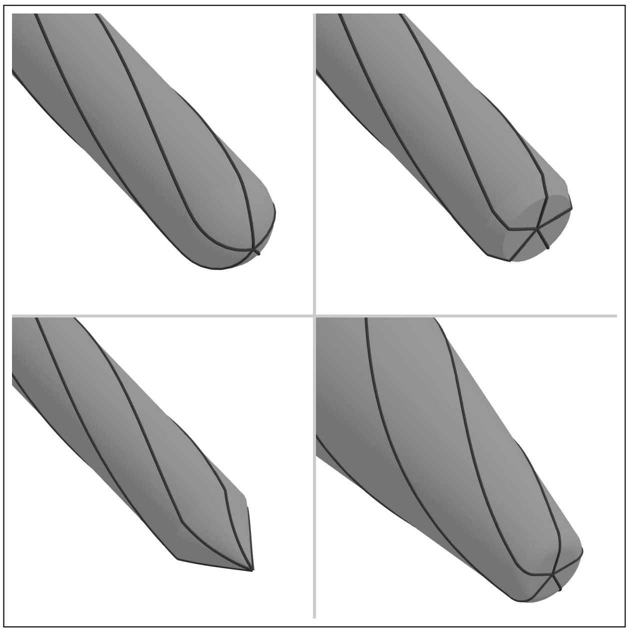
Figure 30 – Rotated points on tool's surface.



Source: Author's archive (2023).

In the final step, the design of the cutting edge was accomplished using the centripetal Catmull-Rom spline algorithm, that generates an interpolating spline that smoothly passes through all the previously rotated surface points. It is integrated into the Three.js library and its implementation is shown in Figure B.1. Figure 31 illustrates different cutting edge designs on multiple tool shapes.

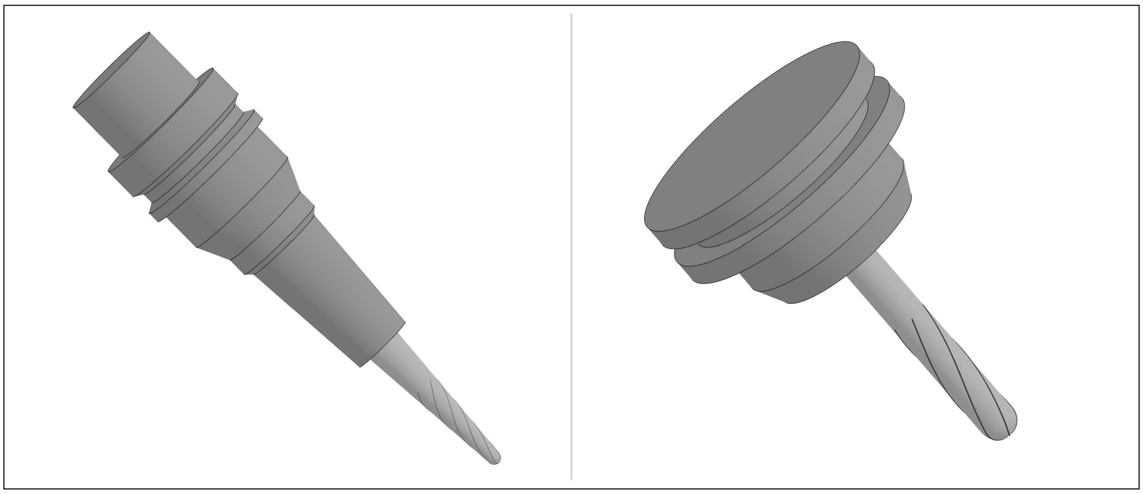
Figure 31 – Cutting edges design on different types of cutting tools.



Source: Author's archive (2023).

The combination of these features enables users to design complex and accurate tool shapes, exemplified by the ones illustrated in Figure 32.

Figure 32 – Different tool shapes designed with the gemineers software.



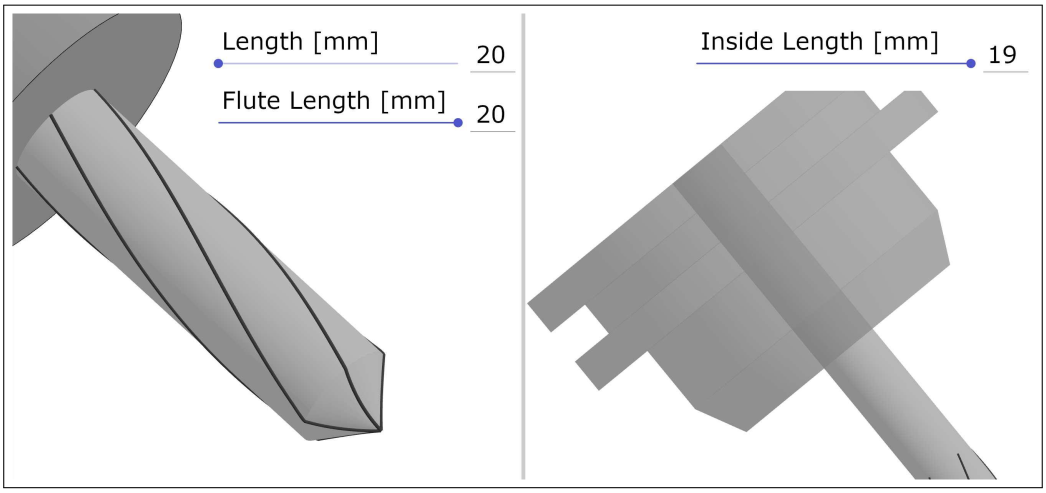
Source: Author's archive (2023).

5.2.3 Results

All interfaces described in this section functioned as intended and were successfully connected to the software's database. They also incorporated several UX Design principles to enhance accessibility and intuitiveness. For example, the Tools Main Page employs a grid layout (optimized for touch-screen devices), as illustrated in Figure A.2 and demonstrates responsiveness, adapting to smaller screens, as demonstrated in Figure A.3. The page also follows the Consistency and Standards UX principle by utilizing industry-standard icons and colors.

The Tool Definition Page was designed to prevent user input errors by dynamically adjusting the allowed range of values for each field based on changes in other fields. For instance, the flute length of a tool cannot exceed the full length of the tool. Similarly, the inside length of the tool cannot be greater than the tool holder length, as illustrated in Figure 33. In this way, the Tool Definition Page not only anticipates errors but also guides users in inputting correct information.

Figure 33 – Dynamic adjustment of input values on Tool Definition Page.



Source: Author’s archive (2023).

The structuring of the tool holder using sections was very important in enhancing the accuracy of tool deflection calculations. The previous method made assumptions about holder geometries that were considerably distant from reality, leading to inaccurate deflection estimations that heavily relied on fine-tuning for reasonable approximations. The introduction of sections provided a more realistic representation of the tool holder’s geometry, addressing the limitations of the previous approach. While a rigorous comparison between the old and new methods was not conducted, the results obtained strongly indicate a significant improvement of the deflection estimations.

6 CONCLUSIONS

This document contains some of the work developed at an internship at gemineers GmbH focused on enhancing the capabilities of the company's machining quality assessment platform through the integration of a cutting tool monitoring system.

In the realm of machining, diverse industries such as aerospace, automotive, and electronics share a common pursuit of precision and efficiency in their manufacturing processes. The demand for exacting precision arises from the fact that even minor deviations can significantly impact the quality of the final product. Given the substantial costs associated with cutting tools, which can exceed one third of unit production costs, and their direct correlation to product quality and precision, the need for a sophisticated tool management and monitoring system becomes evident.

The motivation of this project was to address that need, providing a way to optimize tool usage, prevent their premature replacement and using beyond their lifetime, and finally leading to a greater efficiency of production processes and reduction of costs.

The developed system's infrastructure makes it available to users a comprehensive set of features, allowing them to view, create, edit, and delete cutting tools within the **gemineers** software. One notable feature is the integration of a 3D viewer, providing users with a real-time representation of the tool's appearance as they create or edit it. This not only enhances user experience but also serves as a safeguard against errors. Given the importance of tool geometry in wear estimation, this approach ensures that the system's output is not compromised by inaccuracies introduced during tool creation or modification.

The lifetime monitoring solutions integrated into the system represent a paradigm shift in assessing tool wear. One approach consists in tool usage analysis by each surface point of the tool, offering a better understanding of wear patterns and effectively extending the lifetime of the cutting tool. The other leverages advanced data analytics, incorporating metrics such as volume removed and spindle load to estimate the wear stage of the tool. These approaches provide an insight into the tool's lifespan, allowing for a more proactive tool replacement strategy.

In conclusion, the objectives initially set for this project have been successfully realized. The implementation of a tool management and monitoring system within the gemineers software has met the initial expectations. The system's user-friendly interface, along with advanced features such as the 3D viewer and lifetime monitoring solutions, offers innovative monitoring capabilities to the user and places gemineers in a notable position within the machining technology landscape.

6.1 FUTURE WORK

In terms of the system's infrastructure, there are opportunities to improve the application's usability. Introducing a tool duplication feature would simplify the process of defining similar tools by eliminating the need to manually input each parameter in the Tool Definition Page. Additionally, incorporating a functionality that allows users to save a set of tool parameters as a model could further facilitate the creation of new tools by enabling the loading of predefined configurations.

Turning to the lifetime monitoring solutions, it is necessary to validate the Surface Point Analysis hypothesis through practical testing with real tools. Subjecting these tools to rigorous testing conditions will enable the measurement of wear, validating the effectiveness of the analysis. Similarly, for the Specific Load Analysis, exploring this method's potential by determining maximum specific load ranges for various tool types is advisable, and storing this information will contribute to a more accurate estimation of replacement strategies for future tools.

REFERENCES

1 VAGNORIUS, Zydrunas; RAUSAND, Marvin; SØRBY, Knut. Determining optimal replacement time for metal cutting tools. **European Journal of Operational Research**, Elsevier, v. 206, n. 2, p. 407–416, 2010. DOI: 10.1016/J.EJOR.2010.03.023.

2 GEMINEERS GMBH. **gemineers**. Available from: https://en.gemineers.com/. Visited on: 19 Oct. 2023.

3 FRAUNHOFER INSTITUTE OF PRODUCTION TECHNOLOGY. **High performance cutting – Modern machining in the digital age**. Available from: https://www.ipt.fraunhofer.de/en/Profile/departments/high-performance-cutting.html. Visited on: 19 Oct. 2023.

4 FRAUNHOFER INSTITUTE OF PRODUCTION TECHNOLOGY. **Fine machining and optics – Passion for precision**. Available from: https://www.ipt.fraunhofer.de/en/Profile/departments/fine-machining-optics.html. Visited on: 19 Oct. 2023.

5 FRAUNHOFER INSTITUTE OF PRODUCTION TECHNOLOGY. **Fiber-reinforced plastics and laser system technology**. Available from: https://www.ipt.fraunhofer.de/en/Profile/departments/fiber-reinforced-plastics-laser-system-technology.html. Visited on: 19 Oct. 2023.

6 FRAUNHOFER INSTITUTE OF PRODUCTION TECHNOLOGY. **Fraunhofer IPT**. Available from: https://www.ipt.fraunhofer.de/de/profil.html. Visited on: 19 Oct. 2023.

7 RAPIDDIRECT. **Types of Machining Process: Classifications and Differences**. Available from: https://www.rapiddirect.com/blog/types-of-machining-operations/. Visited on: 30 Oct. 2023.

8 KENNAMETAL. **Get Started with End Mills**. Available from: https://www.kennametal.com/us/en/resources/blog/metal-cutting/beginners-guide-to-end-mills.html. Visited on: 31 Oct. 2023.

9 GWS TOOL GROUP BLOG. **The In's a Out's of Ball Nose End Mills**. Available from: `https://www.gwstoolgroup.com/the-ins-and-outs-of-ball-nose-end-mills/`. Visited on: 31 Oct. 2023.

10 HARVEY PERFORMANCE COMPANY. **The Multiple Uses of a Chamfer Mill**. Available from: `https://www.harveyperformance.com/in-the-loupe/multiple-uses-chamfer-mill/`. Visited on: 31 Oct. 2023.

11 PYLE, Tom. **How To Increase Productivity with Tapered End Mills - In The Loupe**. Available from: `https://www.harveyperformance.com/in-the-loupe/tapered-end-mills/`. Visited on: 31 Oct. 2023.

12 DOLINŠEK, S.; KOPAC, Janez. Mechanism and types of tool wear; particularities in advanced cutting materials. **Journal of Achievements in Materials and Manufacturing Engineering**, v. 19, Sept. 2006.

13 BEHERA, Bikash Chandra; CHETAN; GHOSH, Sudarsan; RAO, P. Venkateswara. 2 - The underlying mechanisms of coolant contribution in the machining process. In: PRAMANIK, Alokesh (Ed.). **Machining and Tribology**. [S.l.]: Elsevier, 2022. P. 37–66. ISBN 978-0-12-819889-6. DOI: `https://doi.org/10.1016/B978-0-12-819889-6.00003-4`.

14 ALHADEFF, Lisa; MARSHALL, M.; SLATTER, Tom. The influence of tool coating on the length of the normal operating region (steady-state wear) for micro end mills. **Precision Engineering**, v. 60, p. 306–319, July 2019. DOI: `10.1016/j.precisioneng.2019.07.018`.

15 UMARAS, Eduardo; BARARI, Ahmad; TSUZUKI, Marcos. Intelligent Design Tolerance Allocation for Optimum Adaptability to Manufacturing Using a Monte Carlo Approach. **IFAC-PapersOnLine**, v. 52, p. 165–170, Jan. 2019. DOI: `10.1016/j.ifacol.2019.10.017`.

16 GARCÍA-MARTÍNEZ, Enrique; MIGUEL, Valentín; MARTÍNEZ-MARTÍNEZ, Alberto; AYLLON PEREZ, Jorge. A new model to predict the tool life in turning of titanium aluminides. **The International Journal of Advanced Manufacturing Technology**, v. 125, p. 1–14, Feb. 2023. DOI: `10.1007/s00170-023-11090-0`.

17 DUONG, Xuan Truong; MAYER, Rene; BALAZINSKI, Marek. Initial tool wear behavior during machining of titanium metal matrix composite (TiMMCs). **International Journal of Refractory Metals and Hard Materials**, v. 60, July 2016. DOI: 10.1016/j.ijrmhm.2016.07.021.

18 HAKIM, M.A.; ABAD, Manuel; ABDELHAMEED, M.M.; SHALABY, Mohamed; VELDHUIS, S.C. Wear behavior of some cutting tool materials in hard turning of HSS. **Tribology International**, v. 44, p. 1174–1181, Sept. 2011. DOI: 10.1016/j.triboint.2011.05.018.

19 MCKINSEY & COMPANY. **What is digital-twin technology?** Available from: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology. Visited on: 2 Nov. 2023.

20 IBM. **What is a digital twin?** Available from: https://www.ibm.com/topics/what-is-a-digital-twin. Visited on: 2 Nov. 2023.

21 IBM. **What is a REST API?** Available from: https://www.ibm.com/topics/rest-apis. Visited on: 2 Nov. 2023.

22 MDN WEB DOCS. **An overview of HTTP**. Available from: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview. Visited on: 12 Nov. 2023.

23 SCHADE, Amy. **Responsive Web Design (RWD) and User Experience**. Available from: https://www.nngroup.com/articles/responsive-web-design-definition/. Visited on: 12 Nov. 2023.

24 KALEY, Anna. **Match Between the System and the Real World (Usability Heuristic #2)**. Available from: https://www.nngroup.com/articles/match-system-real-world/. Visited on: 12 Nov. 2023.

25 SHERWIN, Katie. **Natural Mappings and Stimulus-Response Compatibility in User Interface Design**. Available from: https://www.nngroup.com/articles/natural-mappings/. Visited on: 12 Nov. 2023.

26 KRAUSE, Rachel. **Maintain Consistency and Adhere to Standards (Usability Heuristic #4)**. Available from: https://www.nngroup.com/articles/consistency-and-standards/. Visited on: 12 Nov. 2023.

27 LAUBHEIMER, Page. **Preventing User Errors: Avoiding Unconscious Slips**. Available from: https://www.nngroup.com/articles/slips/. Visited on: 12 Nov. 2023.

28 ATLASSIAN. **What is Git?** Available from: https://www.atlassian.com/git/tutorials/what-is-git. Visited on: 20 Nov. 2023.

29 ATLASSIAN. **Gitflow workflow**. Available from: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow. Visited on: 20 Nov. 2023.

30 VISUAL PARADIGM. **What is Unified Modeling Language (UML)?** Available from: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/. Visited on: 20 Nov. 2023.

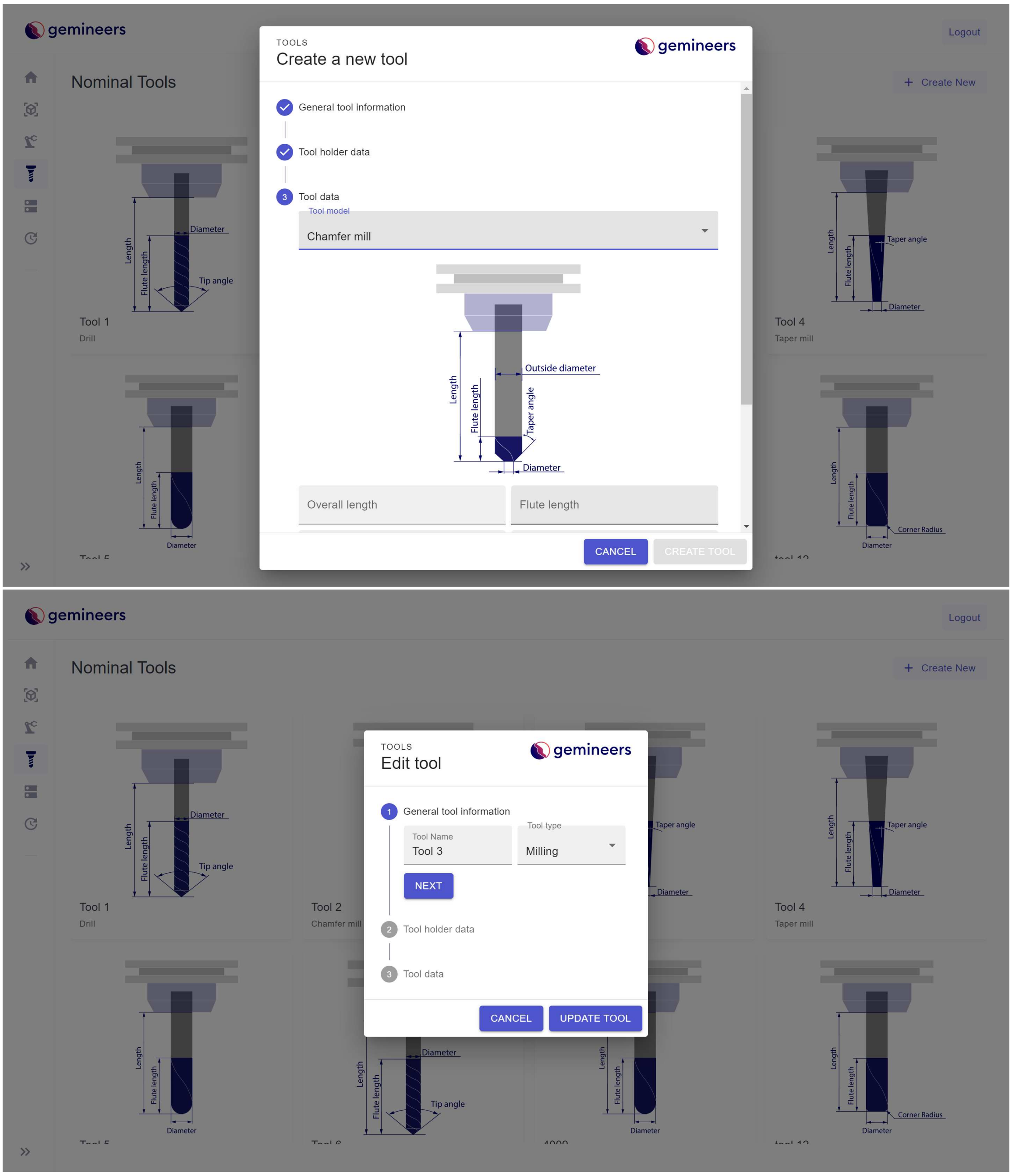
31 MILLINGTON, Sam. **A Solid Guide to SOLID Principles**. Available from: https://www.baeldung.com/solid-principles. Visited on: 20 Nov. 2023.

32 LAMELA, Joaquin. **Applying SOLID Principles in React**. Available from: https://www.octobot.io/blog/solid-principles-react/. Visited on: 20 Nov. 2023.

33 STEPHAN, Abbey. **What Is A Modal in Web Design? + How To Use One**. Available from: https://www.webfx.com/blog/web-design/examples-of-modal-windows/. Visited on: 29 Nov. 2023.

APPENDIX A – ADDITIONAL SCREEN SHOTS OF THE SOFTWARE

Figure A.1 – Tool creation and edition modals in Tool Main Page.

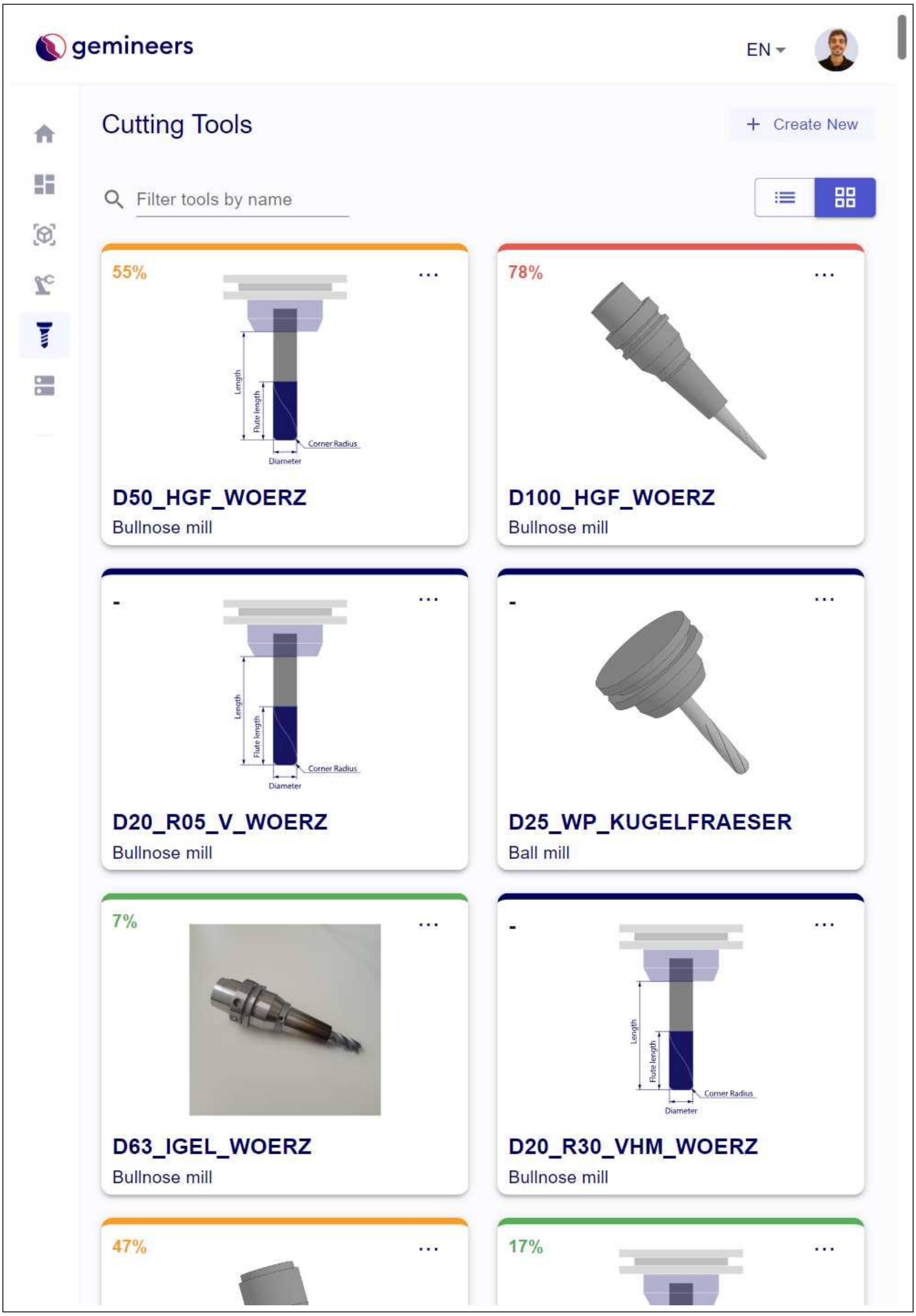


Source: Author’s archive (2023).

Figure A.2 – Tool Main Page in grid layout.

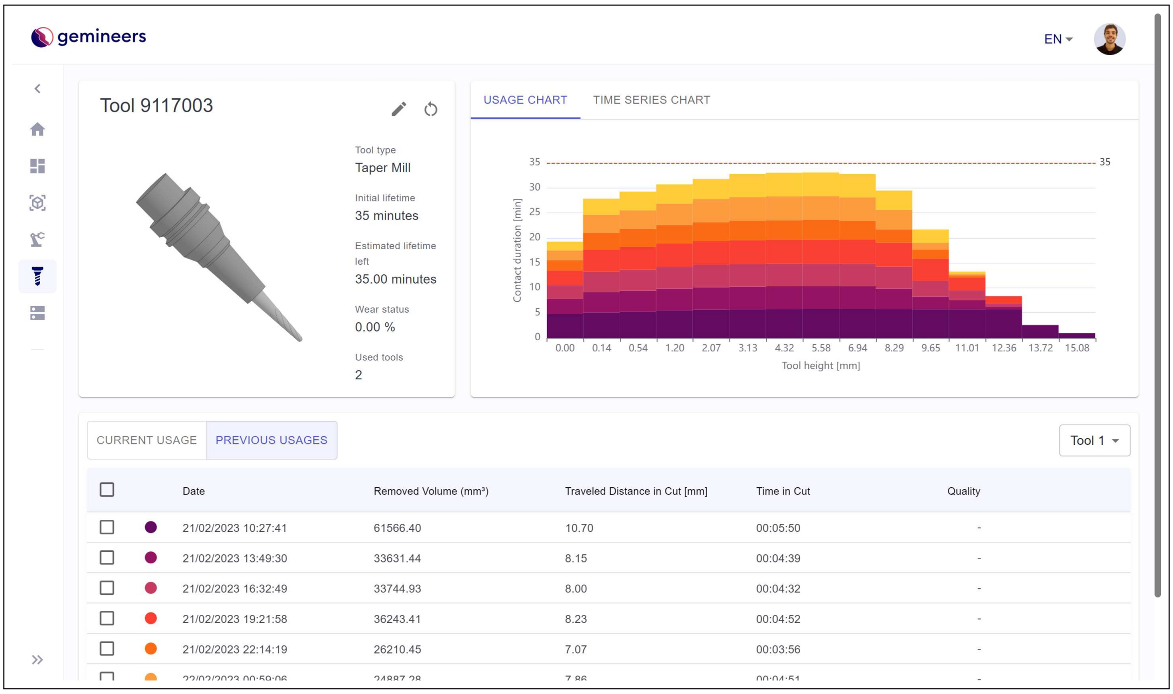
Source: Author's archive (2023).

Figure A.3 – Tools main page on an iPad Air screen.



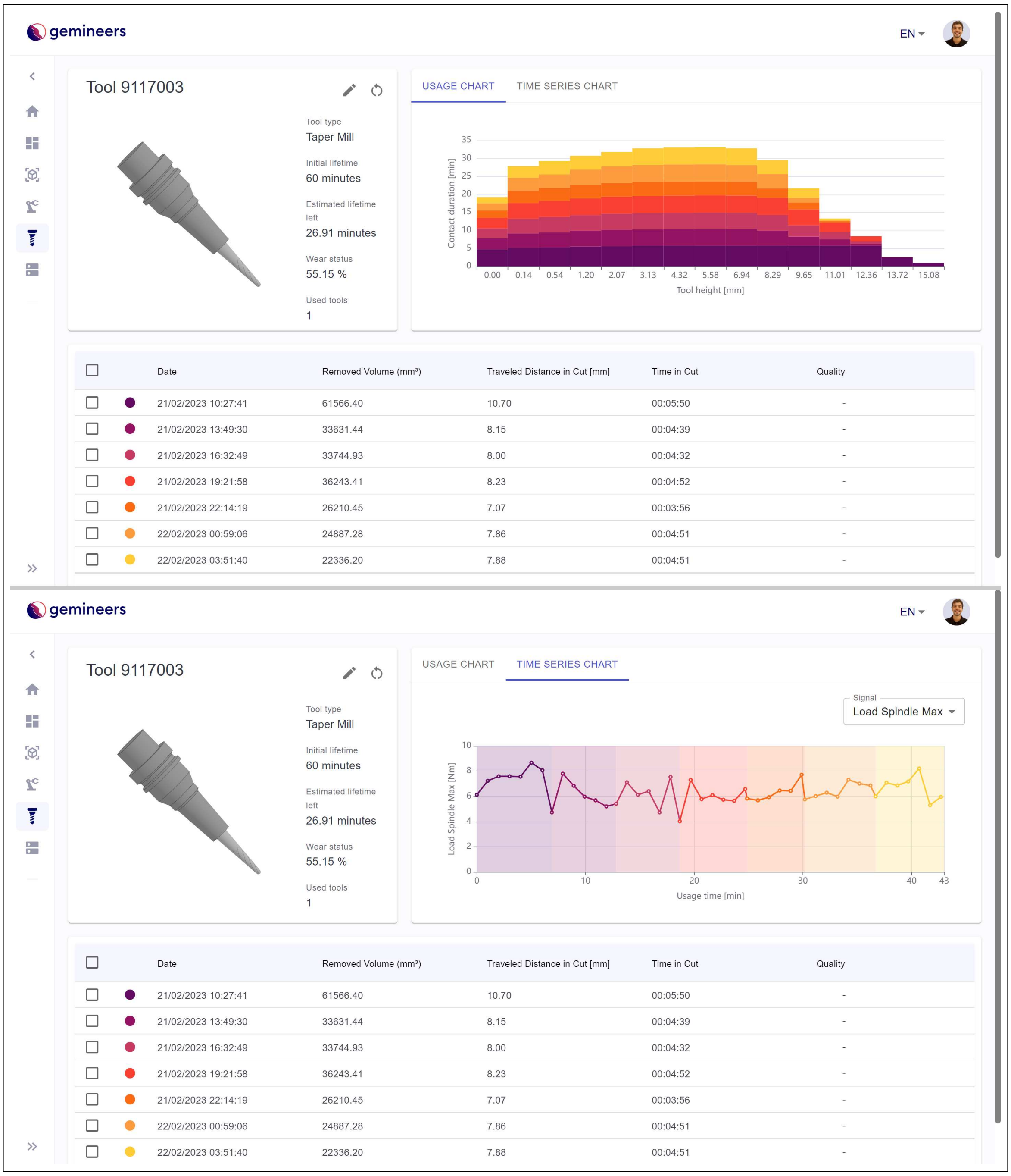
Source: Author's archive (2023).

Figure A.4 – Tool Monitoring Page after tool is reset.



Source: Author's archive (2023).

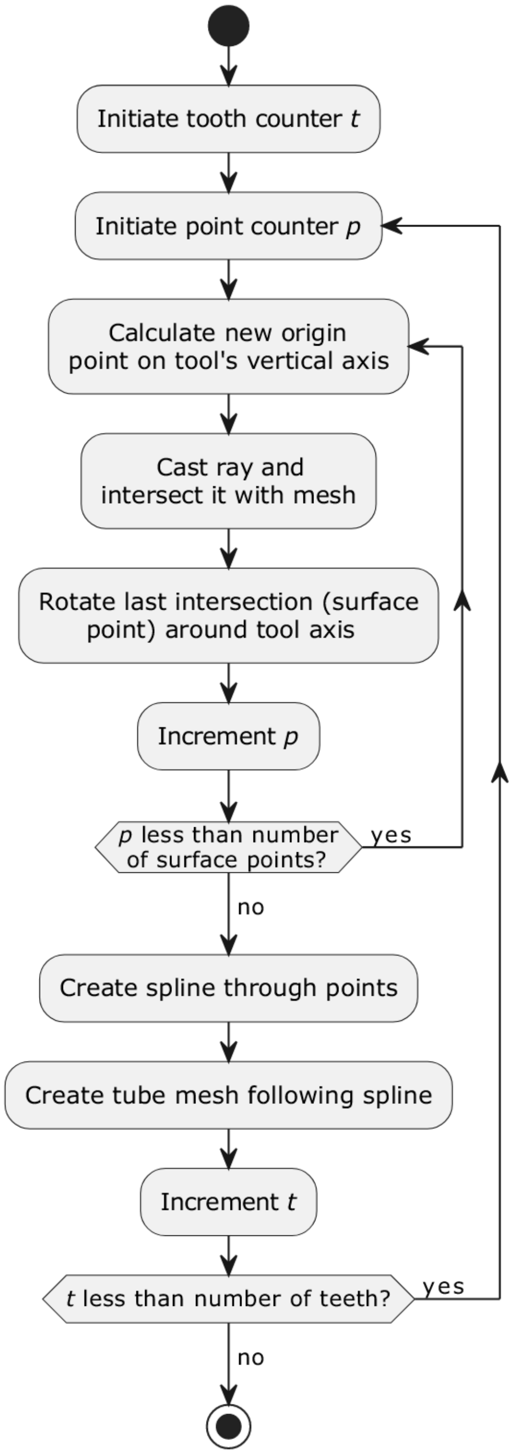
Figure A.5 – Tool Monitoring Page tab feature.



Source: Author's archive (2023).

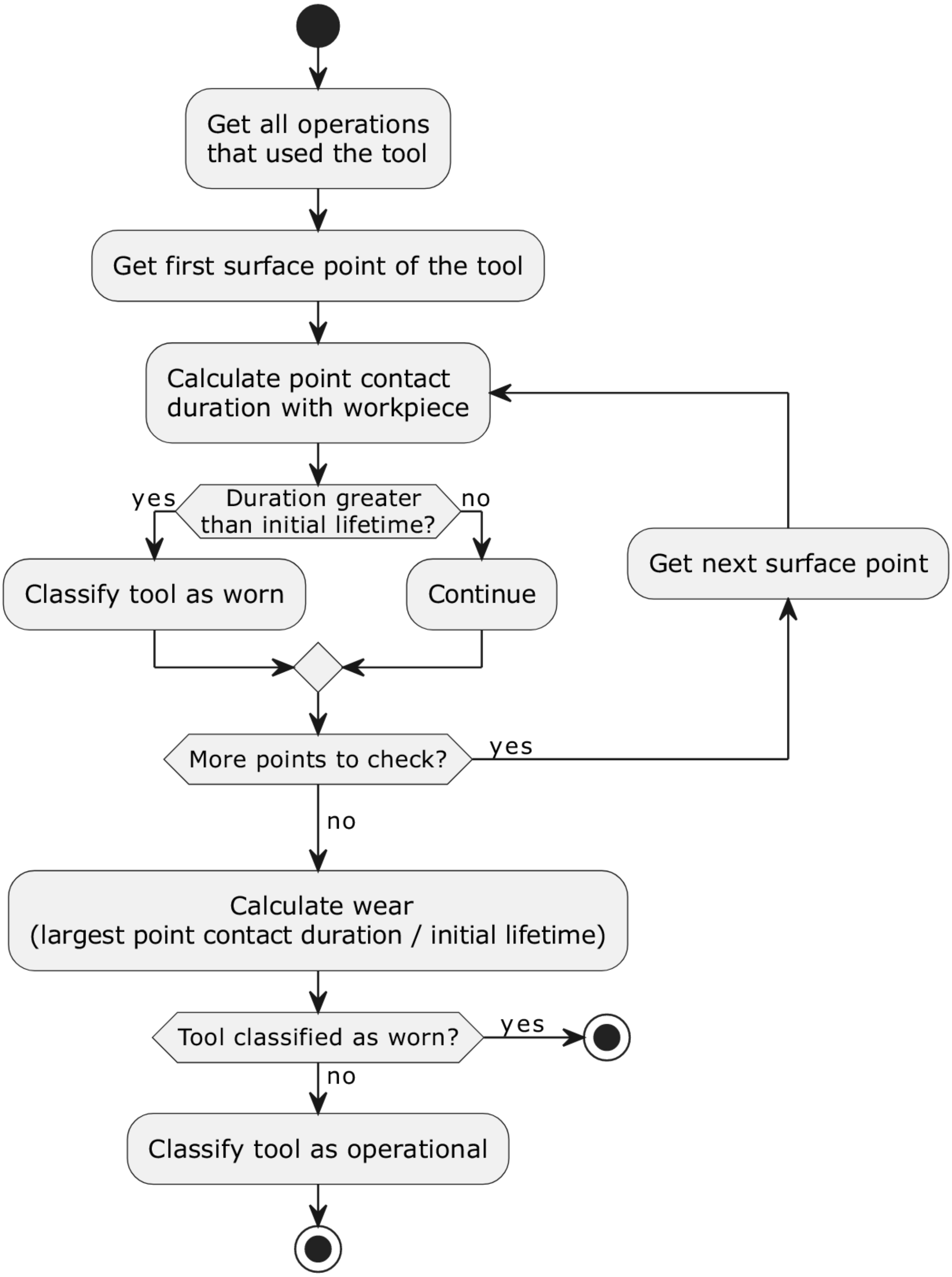
APPENDIX B – UML DIAGRAMS

Figure B.1 – Cutting edges display Activity Diagram.



Source: Author's archive (2023).

Figure B.2 – Wear calculation Activity Diagram.



Source: Author's archive (2023).