

AILIXR

Designing **AI** projects using
a reactive digital canvas

Master Thesis

Ruben Dekker

Integrated
Product Design

Delft University
of Technology,
December 2022



Designing AI projects using a reactive digital canvas

Master Thesis

Integrated Product Design

Delft University of Technology

Faculty of Industrial Design Engineering

Author

Ruben Dekker

Supervisory Team

Alessandro Bozzon, Chair

Phil van Allen, Mentor



PREFACE

During the first years of my studies, I have often wondered whether industrial design engineering has been the right choice for me. Nonetheless, I had not considered any real alternatives, so IDE it was. After a rough start I seemed to get the hang of it however and I have not looked back since. I now feel confident in my design abilities and have enjoyed working on (almost) any project that's been thrown at me. Still, I have never felt like a (stereo)typical designer. My interest in programming has grown at least as much as my interest in design these past few years.

This project allowed me to combine both and has helped me to not feel like only a half designer, half programmer, but both sides fully. Additionally, my interest in educational tools and recent introduction to machine learning meant this project was right up my alley.

Many of the tools used in the final prototype are heavily inspired by the ones found in Nadia Piet's *AI meets Design Toolkit* (2019). Nadia took the time to analyze her toolkit together with me, and this helped guide the project immensely.

To my supervisors, thank you for guiding me through this project. It has had a slow start but the end result is something we can be proud of.

Lastly, I would like to thank my family for their support, not just during this thesis but throughout my entire studies.

RUBEN DEKKER

December 15th 2022, Lepelstraat

EXECUTIVE SUMMARY

Designers have a role in the field of artificial intelligence by creating the interface for the user to interact with the AI. Design is about making things look good', ...that is, according to the latest GPT-3 language generation model¹. In reality, the role of designers within the field of artificial intelligence (AI) and machine learning (ML) is much more influential, especially considering the meteoric developments in AI and its rising influence in our daily lives.

Therefore, supporting designers working with AI is important, yet the tools they have at their disposal are in many ways not satisfactory. The aim of this project was therefore to design a digital canvassing tool which lets designers **explore** AI and ML, **adopt** the affordances that AI and ML as a design material give, and which can be used to **map out** and **design** integrated products, services and systems (iPSSs) by not just guiding the design process but by becoming an integral part of it.

The initial phase of the project saw experimentation with web development technologies and the exploration of the topics of design, design education and AI/ML. Combined with interviews with design students these items formed the solution space from which to start ideating. Out of this phase came three concepts, one of which, *AILIXR*, was to be further developed. *AILIXR* is a highly interactive smart digital design canvas featuring several interactive miniature design tools. Built on the insights that a design-first approach was a necessity, the tool aims to bridge the gap between design and AI by utilizing the design foundation of the students and build towards AI from there. Analysis of other ML design toolkits showed an immense need for extra interactivity. To achieve these goals, staying close to design interactions turned out to be key.

¹ Specifically, the text-davinci-002 model created by OpenAI. The two questions asked were:

Q: What role do designers have in the field of artificial intelligence?

Q: What is design all about?

The application was prototyped, tested and evaluated by design students and experts in the final stage of the project. The final iterations made in response to their feedback resulted in the prototype being ready to be used in projects, as was my personal ambition for this project. Based on the testing feedback and prototyping insights, a roadmap was formed which details the project's future in terms of eventual features and technical improvements.

AILIXR allows designers to explore and adopt the affordances of AI/ML using the highly engaging, interactive and responsive design tools. *AILIXR* can be used as an educational tool by loading in case studies, making it valuable for both the university and its students, or it can be used in real life projects, providing value for design practitioners.

TABLE OF CONTENTS

Preface	4
Executive Summary	5
Table of Contents	8
1. Introduction	12
1.1 Aim of the Project — The Need for Better Tools	13
1.1.1 Project Goals	13
1.1.2 Format of the Toolkit	14
1.1.3 Project Scope	14
1.2 Target Group	15
1.3 Project Stakeholders	15
1.3.1 Design Students	15
1.3.2 The Delft University of Technology	16
1.4 Project Approach	17
1.5 Project Outcomes	18
2. Designing Design Education	19
2.1 Defining Design	20
2.2 Design Education	21
2.2.1 Beginnings of Design Education	21
2.2.2 The Future of Design Education	21
2.3 Affordances of Digital Education	23
2.3.1 Accessibility and Connectivity	23
2.3.2 Interactivity and (a Redefinition of) Reactivity	23
2.3.3 Computational Power	24
2.4 Learning and Design	25
2.4.1 The Myth of Improvement through Technology	25
2.4.2 The Myth of Learning Preferences	26
2.4.3 Theories on Learning	26
2.4.4 Designing (Design) Learning	28
	8
Designing AI projects using a reactive digital canvas	

3. AI & ML	30
3.1 What is AI?	31
3.2 Types of Machine Learning Techniques	31
3.2.1 Classification	31
3.2.2 Regression	32
3.2.3 Clustering	32
3.3 The ML Process	32
3.3.1 Acquiring Data	33
3.3.2 Cleaning the Data	33
3.3.3 Choosing an Algorithm and Training a Model	33
3.3.4 Evaluating the Model	34
3.3.5 Using the Model	34
3.4 Relation to Design	35
3.4.1 AI as the Designer	35
3.4.2 Designers in AI	36
3.4.3 Interviews with Design Students Studying ML	36
4. Ideation	38
4.1 Ideation Foundation	39
4.1.1 Framing the Tool's Approach — ML into Design or Design into ML?	39
4.1.2 Lessons from Other ML Design Tools	40
4.1.3 Increasing Interactivity Through Design	41
4.1.4 Instructional Versus Tool-like	42
4.2 Ideation Methods	44
4.4 Concept 2 - ML Design Boundaries	46
4.5 Concept 3 - AI Filter Design	47
4.6 Concept Choice	48
4.7 Chapter Conclusion	48
5. Prototyping	50
5.1 Determining the Prototyping Goals and Process	51

5.2 Tech Stack	52
5.2.1 Web Development Basics	52
5.2.2 Current Tech Stack	52
5.3 Prototyping Scope	53
5.4 Ensuring Longevity	54
5.4.1 Maintainability	54
5.4.2 Modularity and Expandability	55
5.5 Prototyping Methods and Phases	55
5.5.1 Kanban	55
5.5.2 Agile Development	56
5.5.3 Prototyping Sprints	57
5.6 Demos to ML Experts	57
5.6.1 Demo Setup	57
5.6.2 Demo Procedure	57
5.6.3 Demo Outcomes	58
5.6.4 Changes Made Following the Demo	61
6. Final Design	62
6.1 AILIXR Overview	63
6.1.1 AILIXR Overview	63
6.1.2 User Interface	63
6.1.3 General Controls	67
6.1.4 Complete Website Overview	68
6.2 General Canvas Items	69
6.2.1 Notes	69
6.2.2 Images	69
6.2.3 UX Challenges	69
6.2.4 Frames	70
6.2.5 Icons	70
6.3 ML Widgets	71
	10

6.3.1 Templates and Matrices	72
6.3.2 Trade-off Sliders	77
6.3.3 Benchmarking	79
6.3.5 Data Oriented Widgets	81
7. Concept Validation	83
7.1 Research Questions	84
7.2 User Test Setup	84
7.3 Testing Procedure	84
7.4 Analysis and Results	87
7.5 Testing Limitations	89
8. Conclusion	90
8.1 Meeting Design Goals	91
8.2 Roadmap of the Project's Future / Future Challenges	92
8.2.1 The Ideal Design	92
8.2.2 Technical Roadmap	92
8.2.3 Feature Roadmap	93
References	95
Appendices	100
Appendix A: Project Brief	100
Appendix B: Feedback from the Demos given to ML/Design Experts	101
Appendix C - Future Tech Stack	104
Appendix D – Notes Taken During User Tests	106
Appendix E - System Usability Scale (1-5)	110
Appendix E - System Usability Scale (6-10)	111
Appendix F - AttrakDiff Scale	112

I. INTRODUCTION

This first chapter aims to provide the necessary background information. The original project brief can be found in Appendix A.

1.1 Aim of the Project — The Need for Better Tools

1.1.1 Project Goals

The scope of design is ever increasing, driven by technological and societal changes. Design has grown immensely over the past few decades and has even moved into other fields. Developments in one of these fields, AI, caused arguably the greatest technological revolution in recent times. AI already has a great impact on our daily lives. Therefore, supporting designers working with AI is of great importance. To be able to learn efficiently and work effectively in any discipline, you need the right tools. Yet the tools designers have at their disposal for working with AI are in many ways not satisfactory.

The interest in AI/ML is high, adoption of this design material is however quite low. Tools and materials found online regarding these topics often pertain to how these algorithms are to be implemented. The immense complexity found in AI/ML systems however calls for more than a technical understanding. On the other hand, the AI/ML *design* toolkits which do integrate the design process seem to do so in an abstract and non-interactive sense, thereby not being as engaging and not integrating well into the actual design process. Designers are in need of better tools to work with AI, tools which fit with their discipline and way of working, capabilities and needs.

This graduation project was aimed at creating a toolkit which supports design students (and as a secondary target group design practitioners) in learning how to use AI, more specifically ML, as a design material by means of an online interactive application. Design materials are the tools used in the design process. These can be concrete or abstract, pen and paper can be design materials for instance. In the case of AI, designers can put the possibilities AI gives in their own personal toolbox, their set of design knowledge, skills and tools. When designers feel comfortable designing a smart product, it shows they've added AI to their collection of design materials. For instance, in the case of the design of a smart thermostat, designers can go beyond the simple goal and also argue how it would set the temperature based on data from the user's routines and the weather, and discuss this with ML engineers.

This tool could then be used in both design projects and in an educational setting. An interactive AI/ML design toolkit would help bridge the gap between theory and practice, bringing together a deep abstract understanding of the material and some basic skills on how to implement it.

1.1.2 Format of the Toolkit

From the project's start it was determined the format of the outcome should be web based. Because of the digital nature of the toolkit's subject (AI/ML), it only makes sense for the toolkit itself to be digital as well. Offering the toolkit digitally will offer several significant benefits, some of which will be explored in the following chapter. The eventual design ended up taking the form of an online canvas tool.

The eventual choice for a canvas as the tool's format aligns well with the nature of designing. Canvases offer an engaging and interactive way of working. They give users a better overview of the context, one which they can shape themselves. Similarities to other canvas apps like Miro are therefore recognizable but the final design brings more to the table by being focused on designers and AI.

Although the initial project brief set the project's general direction by clearly stating the target group and even the format of the outcome of the project as being a web app, there was no clear singular problem to solve. Rather, the initial brief simply desired a general improvement of the status quo.

Designing for improvements however tends to lead to incremental change which is not the sort of result I had in mind. Finding clearer problems and their roots generally tends to lead to more radical change. I therefore made the decision to not start the project with the analysis of ML design tools but instead to look at design, education, AI/ML and their relation before diving into existing tools. The following two chapters document this exploration while an analysis of design tools was only done right before the ideation phase.

1.1.3 Project Scope

A major personal ambition for this project was to deliver not just a theoretical design but a functional product or prototype which can be meaningfully used by students upon the project's conclusion. This does not mean the product has to be finished completely, or even polished for that matter, but there should be enough to get valuable use out of. The reasons for this aim are twofold: on the one hand there is the disappointing reality that many graduation projects are not continued or built upon which, depending on the project's outcome, might be a shame. The second driving factor was my own personal goal of learning more about developing web applications.

1.2 Target Group

The project brief clearly states the target group for which to design: design students. A personal ambition is naturally to design the tool in such a way that it can be used meaningfully by many people. Losing focus and being too generic would undermine this desire completely. The opposite, focusing too much on the students taking the ML4Design course at the Delft University of Technology, would however render it useless to outsiders.

Any person seeking to use the affordances of ML to create value for people, business or society, is a designer at that moment and therefore belongs to the target group. This difference between designer and design student is smaller than it appears and mainly comes down to the fact that design students specifically are available to conduct user tests with and will utilize the tool during the ML4Design course. Going forward, designers aiming to employ the affordances of ML should be seen as the target group with design students as the testing group. Differences between both groups are not clear cut and disparity in the design expertise of users should ultimately be kept in mind.

Lastly, since the toolkit will be available to all, by virtue of being web based, it is highly likely that people outside of the target group will also visit the site. Even though the main tool will not be particularly meaningful for them, their possible presence could be kept in mind during the final stages of the project when the tool is made ready to go live.

1.3 Project Stakeholders

At this point there are no external stakeholders/companies connected to this project, though possible future partnerships when continuing the project are naturally not out of the question.

1.3.1 Design Students

The design students of the course Machine Learning for Design will be the primary group of users. This new elective course is given in the second year of the Industrial Design Engineering bachelor at the Delft University of Technology. The students taking this course are expected to have affinity with the topic since the course is an elective, yet most will not possess any prior experience. They want to learn how to be able to envision and design iPSSs (intelligent products, services, and systems) as they see the landscape of design and the world around them changing.

1.3.2 The Delft University of Technology

Making the tool open to others aligns well with the goals of the second main stakeholder, the Delft University of Technology. To start, faculty members of the Industrial Design Engineering department can use the outcomes of this project to supplement courses and learning material for students to help them explore the topics in a more fun and interactive manner. This fulfills part of the university's mission of educating new generations of engineers. The second part is to perform and share valuable research. Making the toolkit available to people outside the university would align with the university's mission of sharing knowledge and research to improve the world around us since an AI/ML design toolkit would enable designers to create better design AI/ML solutions.

1.4 Project Approach

As with most projects, the initial planning has changed significantly compared to the final process because projects tend to shape themselves. Though it was not explicitly planned to follow this method, looking back, the *Double Diamond design process* (Design Council UK, 2005) aligns well with the course the project took. The initial divergent phase of discovery broadly looked at the needs and possibilities the context presented. Here, the main topics of Design and AI/ML were analyzed through interviews, papers and researching other design tools. Valuable insights from this stage were bundled and laid the foundation for the following ideation phase. During the ideation phase three concepts were generated. At the halfway mark of the project, one of these three was chosen to continue with, this concept was then detailed iteratively through prototyping in the final stage. At the end of the project, a user test provided relevant insights for the possible continuation of the project beyond this graduation project which are detailed in the future roadmap in Chapter 8.

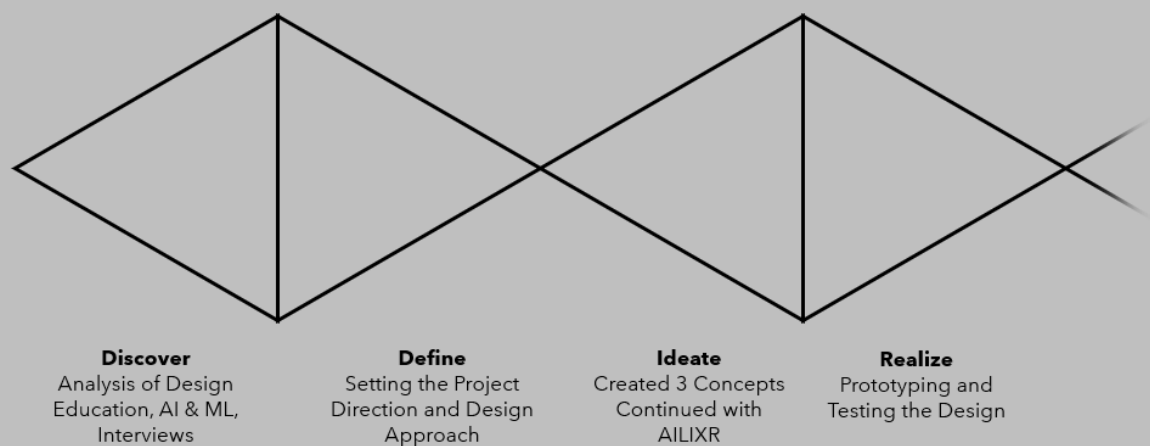


Figure 1: The project's approach roughly followed the Double Diamond design model. Of course, not all activities undertaken during the project fit within the model.

This report follows the project relatively closely. Some stages are not covered in much detail or at all. Quite a sizable part of the project was for instance spent learning technical skills which were necessary for prototyping the final design.

1.5 Project Outcomes

The open-ended outset of the project provided many possible paths. However, these abstract beginnings have eventually led to a concrete design, one which can already be used by students. This main result, *AILIXR*, allows designers to **explore** AI and ML, **learn** about the affordances that AI and ML as a design material give and can be used to map out and **design** integrated products, services, and systems (iPSSs) by not just guiding the design process but by becoming an integral part of it. The highly interactive and reactive canvas design students or professionals create by using the *AILIXR* can be effortlessly shared with others to easily communicate ideas within teams.

2. **DESIGNING DESIGN EDUCATION**

Though Machine Learning is stated plainly in the title of this thesis, the decision to lead with the topic of Design Education when covering fundamental themes, is deliberate. The question of how to teach is just as, if not more, significant than the issue of determining what to teach.

2.1 Defining Design

Before conveying knowledge or a skill, the first step is knowing which knowledge or skills the students already possess. The students in question are familiar with design. But what is design? To many, design is somewhat unknown and associations which run deeper than ‘designers make things look nice’ are not too common. Though the statement above is not explicitly wrong, it is severely lacking. Seeing as this report not only documents the outcomes of design but builds on design as a skill and a profession, the need to address this topic is clear. The unfamiliarity with design as a field which goes beyond creating aesthetic experiences is in part due to the fact that design is not stand-alone but has to be seen in relation to something else. When there are no limits to what this relation can be it follows that the field of design is immensely broad. The Delft University of Technology describes their Industrial Design Engineering programme as follows: “Matching the evolution of people with the revolution of technology. [...] designers act as a bridge between advancements in technology and the needs of people, organizations and society.” (Delft University of Technology, 2022).

This answer is satisfactory when looking at the field of design (design is broad, expanding and applicable to almost anything), yet it tells us nothing on what design is as a skill or process. When asking designers how it is they work, even expert designers have trouble explaining what it is they do exactly (Dreyfus & Dreyfus, 2005), not in the least because creativity is integral to this process (Dorst & Cross, 2001). We must therefore limit ourselves to understanding which competencies designers ought to have. The renewed curriculum for the programme in Delft (Voûte et al., 2020) describes the following five core competencies:

1. Framing & reframing the design challenge in its emerging future context
2. Creating & evaluating iteratively to converge towards a desired impact
3. Integrating an increasing amount of relevant perspectives into a working whole
4. Meaningfully steering the design and stakeholder process
5. Working and communicating at varying and multiple levels of abstraction, and across disciplinary perspectives

The items listed here are by no means an exhaustive collection of what it is exactly that designers have to master, for non-designers it should however paint a picture of what design is about, clearly showing design is more than they would initially expect. The items listed here are still abstract but that is due to its ability to be connected to any other topic. Some go as far as saying design is one of the highest cognitive abilities human beings can possess (Cross, 1996). The key takeaway here being the word *ability*, design is not just a noun, it is in fact a verb.

2.2 Design Education

It is therefore no surprise that design education around the world revolves around practice. The fact that design is an expertise which can only be learned through practice is not surprising, though it is helpful to keep this in mind. The design field is ever evolving and since it is our goal to create a future proof tool, we need to take a step back and see where design education came from to see where it is headed.

2.2.1 Beginnings of Design Education

The breadth of the design field as described in the first section of this chapter was not always there. In fact, Industrial Design as a field of study is quite new. Great economic shifts post the second World War and the ever-increasing impact of mass production saw product design was becoming an established profession. Starting off as an offshoot from the school of architecture in the sixties, the Industrial Design Engineering department in Delft followed a pattern seen all over the world (TU Delft, n.d.-b). Although at the beginning influences from architecture were present, programs would quickly find their own identity. It is interesting to note that even though the focus of the programme was on product design, since the beginning faculty and lecturers came from a wide variety of backgrounds. Hints of the wide reach design would attain were therefore already visible when design education was in its infancy, further demonstrating that design can relate to anything. Originally being limited to designing physical products, the programme has since mirrored the changes which have transformed the profession the world over. Throughout the years, the department grew and many more domains such as AI would find their place within IDE. The goal of educating students to be all-round designers was still there, it became clear however that the conditions to meet said goal would continue to change at a rapid pace.

The latest additions to the programme have mostly been due to technological advancements after the turn of the millennium. The affordances these changes have brought will be discussed in Section 2.3. Some of the most major additions are interface design, interaction design, experience design, service design and systemic design. All the while, the steady growth of IDE as a (scientific) field has continued.

2.2.2 The Future of Design Education

Predicting the future is not an easy task and perhaps this section's title could have been more tentative, but when looking at the history of IDE, a clear trend can already be identified: the scope of design is ever increasing and is driven by technological and societal challenges.

When taking a closer look at this growth, two different types are visible. It should be noted however that these categories are not a simple dichotomy but new topics will lie somewhere between these extremes.

Interface design, interaction design, experience design etc. are all in a sense an addition to design itself. They are, just like the following category, driven by changes in the world, but are also part of design through and through, hence the 'design' suffix.

The second type of growth is an expansion into existing fields, not taking over and claiming it as its own, but the building of bridges. At one such topic which design is seeping into, the combination of both will be the focus of the following chapter. The question of whether the all-rounded designer should have a strong foundation in design itself, and therefore possess the ability to carry their skills over into another environment, or whether they should focus on possessing the knowledge about any bordering areas is an interesting one. At the rate the design field is growing however, the second perspective would become impossible to successfully achieve. The direction design education is taking is not strictly one or the other but the approach of the future seems to be geared towards providing students multiple paths early on, thereby in a sense moving away from the initial goal of shaping an all-rounded designer.

2.3 Affordances of Digital Education

As discussed in the previous section, the curriculum reflects the changes in the world around us. The advent of personal computers and the internet have brought major changes to all facets of life, education included. The effect on design education can be seen all too well when considering the workbenches which were once so prominently placed in the middle of the IDE hall have now been replaced by desks and laptops. Since the tool will be digital, this section discusses some of the affordances of digitalization relevant to this project and is therefore not at all exhaustive.

2.3.1 Accessibility and Connectivity

Accessibility is naturally listed first, after all, having access to something is the first step to benefiting from it. Compared to physical products, digital products can be made once and distributed virtually infinitely. Distribution channels have since shifted from those of conventional physical products to the internet, thereby not only bringing down costs further but also cutting back on time. The logical consequence is a high degree of connectivity, any device with an internet connection (and a browser) can participate. The unsurprising effect is the highly accessible and connected nature of (most) digital products and services. Having instant access to information from anywhere in the world has admittedly been a greater factor than the personal computer itself.

The points in this section are not at all specific to design education, far from it. Still, actively trying to reap full benefits from all these advantages is key, though most will likely be inherently present in the final design.

2.3.2 Interactivity and (a Redefinition of) Reactivity

Both interactivity and reactivity are some of the general benefits that resulted from digitalization. Both are general, yet compared to the previous section, their relation to design is much stronger. Let us first establish a definition for both concepts. Within web design, interactivity is anything within the design which responds to user actions. Reactivity, when related to websites, deals with responding not to user actions, but to changes on the server side of things. The new state (data) is reflected automatically on the website ensuring information is up to date without having to reload the entire page.

These definitions are correct, however, to me, interactivity and reactivity when confined to user interaction and experience in regard to the design take on a slightly different meaning. In effect, the definition of interactivity as stated above can be split in two:

Interactivity is still the response to user actions, through an interface or the design in general. The definition of reactivity as used in web development circles is dropped completely. For the rest of this report, when discussing reactivity, the definition will be the following (unless stated otherwise): reactivity can be seen as a type of, and an addition to interactivity. It is the response to a specific type of user action, a decision. In both instances the system is able to give feedback and the difference between the two might thus seem insignificant, ultimately, in a certain sense almost every action is preceded by a decision. There is however value to be found in this redefinition. A decision is more specific than an action, it follows that the reaction to a decision is too. This redefinition is to say that going beyond designing a system which responds to action, a system which responds to actions and decisions offers users a more complete interaction in the form of custom feedback.

These tangent forms the basis for some of the ideation in the later chapters but when connecting these redefined concepts back to design education the link should become clearer. Designers heavily rely on feedback; this feedback can come in many forms. The most important source is neither completely internal nor completely external. The design itself feeds back and is subsequently changed, it is a dialogue (Cross,1996). Iteration is the basis of design and relies on feedback and any source of feedback is valuable. The interactivity and reactivity digital systems provide can enhance this process immensely by letting designers use smaller and smaller feedback loops. Be it by drawing, modeling, writing, etc., because ultimately, no design is made perfect in one go.

2.3.3 Computational Power

A last major advantage comes in the form of leveraging computational power. This too boils down to both a reduction of time and money but most important to designers, this massively reduces the size of the feedback loop. A smaller feedback loop means faster iteration. Modeling physics, three dimensional objects or creating ML models requires a great amount of computation yet it has become accessible. Digitalization was (and still is) a technological revolution, so it is not surprising that the prototyping activities seem to be some of the facets most accelerated.

Many other aspects of digitalization can be said to benefit design (education) but these seemed the most important to go over with regard to this project.

2.4 Learning and Design

Having looked at design, design education and the affordances digitalization has brought to these subjects, this section will dive deeper into the learning process and how to effectively design education to form a foundation from which to start the ideation process. When researching education, I discovered some of the beliefs I held regarding education were myths, rectifying these redirected the ideation phase for the better.

2.4.1 The Myth of Improvement through Technology

This entire previous section on affordances of digitalization is not to say that any type of technological innovation necessarily improves the learning (and teaching) process. It simply has the *ability* to improve and enhance education. This distinction might sound pedantic, however, giving more thought to how certain new technologies are integrated can lead to more thoughtful use. In a way this myth comes down to the mental bias of something newer automatically being better. Often, technology can simply be a substitute. Changing to a digital medium might result in a more streamlined process of actions, it does not necessarily equate to a better learning process. Substitution also is the first level in the four stage SAMR framework (Puentedura, 2014).

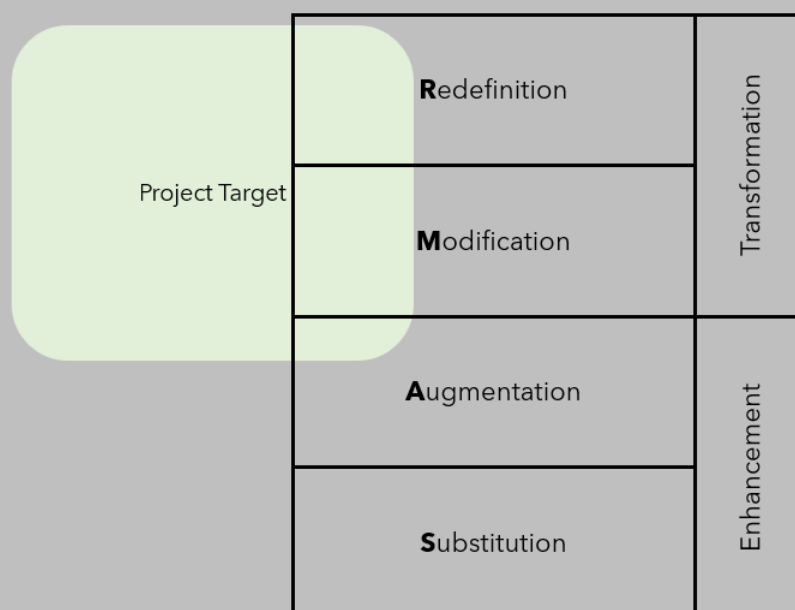


Figure 2: The SAMR framework (Puentedura, 2014) shows that when introducing technology in education you have to be critical.

At this first stage there is no functional change, there is in the following augmentation stage. These first levels show gradations in how an existing method of working can be enhanced in terms of engagement and efficiency. An example of these levels would be the substitution of writing with a pen or using a text editor, this process can be augmented by inserting images in the text. The last two stages do transform the activity by actively changing the learning process. Modification significantly alters the process while the last level, redefinition, allows for a learning process which would have been wholly impossible if not for the new possibilities provided by technological advancements. Comparing a new course, or other types of educational ideas to this framework will force a more critical analysis of whether the technology truly improves the process. The stages to aim for are obviously the latter ones. If a new idea would belong to the first stages a traditional approach is likely to be better since it is possible the technology itself simply detracts and distracts from what you want to convey.

2.4.2 The Myth of Learning Preferences

A more prevalent and alarming myth is the assumption that students perform better when teaching is delivered through a medium which matches their learning preferences. This belief is easy to accept without a second thought, after all, isn't every student unique? No evidence which supports this myth exists, on the contrary, there is proof which plainly contradicts it (Clark, 1982). It is interesting to note that the refutation is just as easy to accept: all students benefit when different learning mediums are combined. A good example is the Drops language learning app: new words are presented to users visually, textually and auditory, all at the same time. However, this title's section is slightly misleading because there is some truth to this myth. Learning styles do exist insofar that students have preferences as to which medium the learning content is presented to them. This preference however does not translate into better learning (Clark, 1982). Focusing on diversifying learning content based on personal preferences is therefore not necessary as was my initial intent, instead the general needs of design students should be looked at.

2.4.3 Theories on Learning

Kolb's Experiential Learning Theory (1984) goes beyond the standard visual, textual, kinesthetic and auditory styles. Instead, the model works with four steps where the focus instead lies on the cognitive process. Learning is seen as the process where knowledge is created through transforming experience. The theory describes learners going through a process for effective learning to take place.

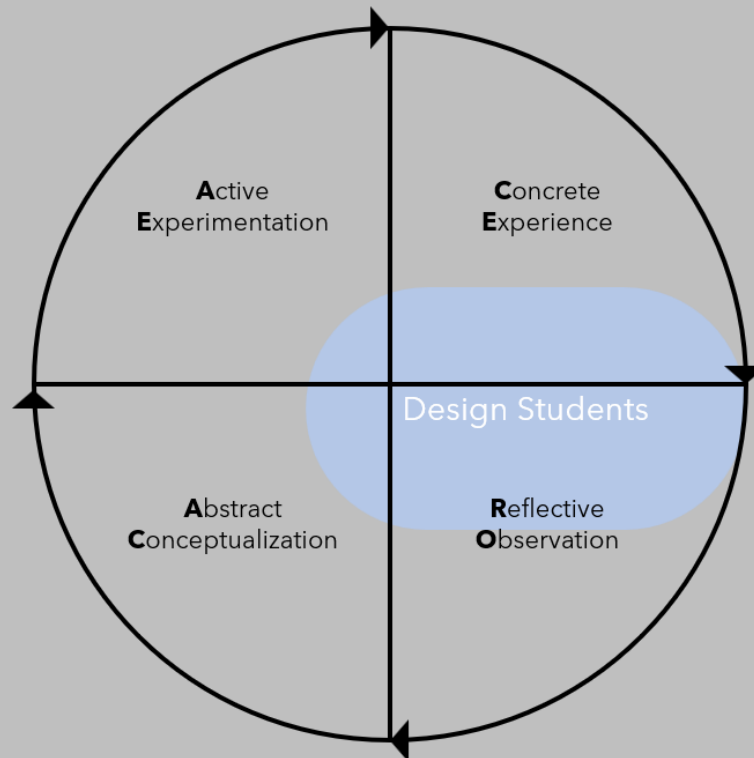


Figure 3: Kolb's Learning Cycle. Design students often have a preference for Concrete Experience and Reflective Observation. All phases are necessary for learning to take place however.

Learners can enter the cyclical learning process at any stage. Diagrams depicting the process show the Concrete Experience (CE) step at the top, followed clockwise by Reflective Observation (RO), Abstract Conceptualization (AC) and Active Experimentation (AE). As discussed in the previous section, different learning styles do exist. People do generally have a preference on which step in the Kolb cycle they prefer. Most IDE students prefer either RO and/or CE, which could be said to follow the applied and reflective nature of design.

A stronger relation to design is perhaps the shape of the process itself. The cyclical nature of learning, according to this and many other theories such as Dewey's (1986), to me strongly resembles the iterative character of the design process. Every design iteration something new is learned after all. According to Donald Schön, designing should be seen as an educational process in its own right as design leads to discoveries which themselves build understanding (1992).

The conclusion is once again that when aiming to teach, the focus should be on creating a learning process which manages to meaningfully combine different learning mediums and (cognitive) processes, all the while achieving a nice flow between them. Building an iterative process into the core of the tool would support the students in their learning and would mesh with design and the design process itself.

2.4.4 Designing (Design) Learning

As has already become apparent from the previous sections, also education can be designed.

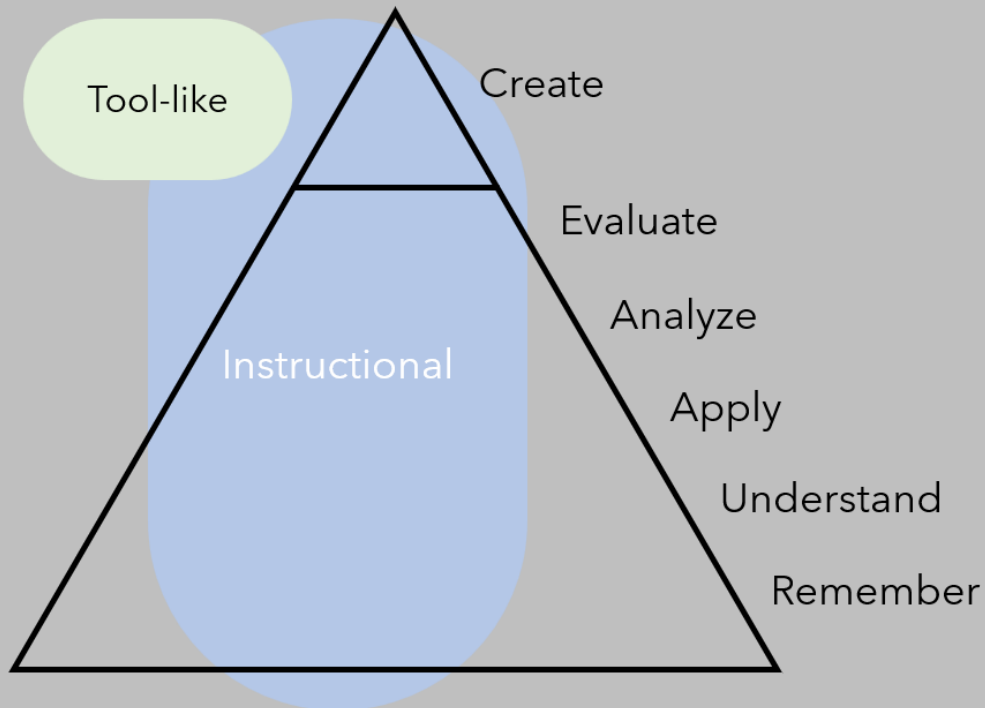


Figure 4: Bloom's Taxonomy of Learning (1956) in relation to this project. The outcome could be used as a creative tool or as a supplement to education. It seems the levels at which students interact with the tool differ but they don't, students simply work from the ground up. Using ML to Create requires all previous levels.

Bloom's Taxonomy (1956) is another more well-known and used pyramid style educational framework, it classifies learning activities based on six different cognitive functions. The remembering of basic facts is again at the foundation. In order the following levels are: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. This final level, Creating, is interesting because it is once again easy to make the link with design. This framework justifies the earlier statement by Cross (1996) that design is one of the highest cognitive abilities. To get to this point however, all preceding levels need to be satisfied. For designers to effectively design for Machine Learning, they need to first be able to know basic facts, understand the concepts, be able to apply this knowledge in a new situation, analyze concepts and evaluate decisions and opinions. Seeing design move into other areas of expertise in this light shows the process to not be as straightforward.

It is therefore also a myth to think knowledge and skills are distinct: skills build on top of knowledge and these both are thus deeply intertwined.

For this reason, it follows that it is also a mistake to think that skills from one domain can easily be transferred into another. Connecting design education to this yields some interesting insights. The definition given in this chapter sees design as a broad ability. Design education then seems to be on the right track by providing a foundation of both knowledge and skills in a wide variety of domains instead of focusing on perfecting core design skills. Designers become part of the system and design context (Casakin, 2006); they cannot simply add their design skills to a new context. Instead, new skills built on knowledge of the new context need to be developed.

3. AI & ML

In addition to the expansions listed in the previous section, another clear trend in the design field is the increase of designing for the digital world. In turn, one of the biggest developments in the digital world has been the advent of Artificial Intelligence (AI).

3.1 What is AI?

The term as we know it today has been conceived in the fifties, yet a single agreed upon definition does not exist since the field has evolved a lot over the years. The enormous influx of attention means that most people are, on a surface level, to some degree familiar with AI. AI is the ability for machines to display intelligence, it is important to note that this does not mean machines mimic human intelligence but rather, “AI is a collection of concepts, problems, and methods for solving them” (Elements of AI, 2022). The second term used in this section’s title is Machine Learning (ML). Often AI and ML are used interchangeably yet this is not accurate, ML is in fact a core subdomain within the AI field. A concise definition is the following: “Giving computers the ability to learn without being explicitly programmed” (Samuel, 1959). Rather than programming an algorithm ourselves, ML aims to solve problems by automatically learning and improving based on data. The algorithm finds patterns; therefore, the quality of the algorithm depends on which data were used to train it. A well-known phrase regarding data within the ML field is “garbage in, garbage out”, that is to say, a model is only as good as the data used to train it.

3.2 Types of Machine Learning Techniques

Some well-known applications of ML include: self-driving cars, recommendation algorithms and speech recognition. The underlying methods driving these smart systems are not the same. ML is in fact an umbrella term which covers three main subdomains: supervised learning, unsupervised learning and reinforcement learning. The basics of the three most common techniques are explained down below.

3.2.1 Classification

Classification models are able to detect languages, predict heart disease, or pedestrian detection in self-driving cars, to name a few examples.

Before explaining Classification, the subdomain of ML to which it belongs needs to be explained. Supervised Learning works using examples. When training a model, the data is supplied with the correct corresponding labels (like an answer sheet). It is now the algorithm’s task to find patterns between data points by looking at the labels.

Classification algorithms are used to create models which can classify new observations into existing classes. A well-trained model can filter into these classes. The labeled input data

determines the classes. Labels might be as simple as 'yes, no', '0, 1, 2' etc. The output of a classification model is not the label itself but a prediction or probability for each class.

3.2.2 Regression

Another type of ML technique within the Supervised Learning subdomain is Regression. The difference with Classification is in the output. Regression is aimed at predicting a numerical value. Technically speaking Classification predicts a probability which is numerical but the goal is to categorize.

Some concrete examples of use cases are the prediction of house prices, someone's age, the weight of an animal and sales predictions.

3.2.3 Clustering

At first glance, Clustering would not seem much different from Classification. Using both techniques, the end result is some sort of categorization. The difference lies in the fact that Clustering is unsupervised, this means that the dataset to train the model is presented to the algorithm without any labels. The algorithm sets out to discover patterns in the data. Clustering does not even have to be the end goal when working in a ML project. Because Clustering is unsupervised (and labels are not necessary), it is an ideal first step to getting a better understanding of the data you have at your disposal and by extension get a better grasp on the context.

Another use for Clustering is generalization; here, items with incomplete feature data can still be grouped by inferring data from the other data points. An example use case is the clustering of new videos with already popular videos to improve recommendations for the YouTube algorithm (Google Developers, 2022).

Specific uses for Clustering include the segmenting of customers for marketing purposes, image compression and clustering books to name a few. Some of the most used algorithms in Clustering are K-Means and DBSCAN.

3.3 The ML Process

The following steps give a general outline of the ML process. As different as the three subdomains from the previous sections are, all follow the same process. It is important to note

that this process is separate from the design process. The exact form of the steps below is not universally agreed upon.

3.3.1 Acquiring Data

As should have become clear from the previous section, the importance of data within the ML process cannot be understated. The rest of the process will fully depend on the amount, quality and type of data you can gather. There are many public data sets available online. Often it is even possible to scrape the data from social media, search engines and other websites. Utilizing user interactions to collect or label data is another possible route. CAPTCHA is a prime example, website visitors are asked to verify they are human by doing a small test, the answers they give are not just for verification but are also used to create data sets. Of course, using real-time data from smart systems or web scraping programs is also a possibility.

In the case the data is unstructured or unlabeled, there are online data preparation services which can help. Lastly, data can also be bought, either as a service or per set. The key to this first step is to be both critical and resourceful.

3.3.2 Cleaning the Data

Being too critical during the data acquisition process is to be avoided however. Most data sets are not immediately usable in models. Common problems are missing data points and unorganized data. Cleaning and preparing the data to make it usable is usually the most time-consuming step in the entire process but also one of the most important: as said in the previous section, poor data leads to poor results. An Excel sheet or a Tweet for instance, cannot be directly used by ML algorithms so data often needs to be translated to a format models can work with such as CSV and JSON.

3.3.3 Choosing an Algorithm and Training a Model

Choosing which ML algorithm to use depends on both the design goal and the data set. Going into more detail on how to choose an algorithm and train a model is best done using a detailed case study but this is beyond the scope of this report.

When working with Supervised Learning algorithms there is an important step to take. The total data set should be split into training and testing data. Often 80% of the data set is used for training.

3.3.4 Evaluating the Model

The ML process as laid out here is not as linear as this list makes it seem. Just like the design process, the evaluation of a model is inherently iterative. Using the testing set from the previous step the performance of the model can be analyzed. Going by accuracy alone is not sufficient, multiple other aspects require careful evaluation. If the results are deemed to be unsatisfactory (which they likely are the first iteration), parameters can be tweaked or a different model can be chosen.

3.3.5 Using the Model

This last step speaks for itself; the model is now ready to be deployed to an application. From a technical standpoint however, deploying a model is not always as straightforward as some models require significant resources to run. Here, there are also services which offer ML deployment. After having deployed the model, it is advisable to reevaluate it after a period of time and collect performance statistics.

3.4 Relation to Design

3.4.1 AI as the Designer

At the time of writing this report, an interesting story broke. Headlines such as ‘AI won an art contest’ (CNN Business, 2022) piqued the interest of many. Setting aside the title implies AI being something possessing agency, thereby violating one of the principles in section 3.1.1, it nevertheless is fascinating news. The relation between AI and design can be explored in multiple ways. For starters, let’s tackle the question of whether AI can design. Donald Schön (1992), when trying to answer this problem, starts by making a distinction between phenomenological and functional equivalence to design. Functional equivalence here is comparing the output of the (human) design process with the output of an AI model. In the article Schön sees no way for a design program to decompose its inputs from the design situation all the way down to prototypical rules and relations.

Given that the article was published in the nineties, a lot has changed since. ‘AI’ winning an art contest is proof, is it not? It would seem so, but there is more to the story. The winner of the art contest was not ‘an AI’, but rather a designer who had used the Midjourney AI which creates images from a textual input. Creating the final result was not as straightforward: over nine hundred iterations and eighty hours of work went into the winning image. This corresponds with Schön's conclusion that AI will likely end up being a valuable assistant to designers, for example by generating simulated environments and extending designers’ prototyping repertoire. Midjourney is a type of generative AI. Generative AI is the term for “programs that can use existing content like text, audio files, or images to create new plausible content” (IBM developer, n.d.), and as such can definitely be used as a design tool.

The answer to whether AI can attain phenomenological equivalence depends whether it is able to mirror the abstract cognitive processes designers go through. Although it is intriguing to envision such a future, it is clear that presently no models come close. The description of the creative process by Schön aligns more closely with that of super intelligence than that of narrow AI. A Turing test for design models is a fascinating idea but whether this distinction between the two types of equivalence matters if the output is indistinguishable is debatable. Ultimately, the technology is not there yet and the designer (still) remains fully in control.

3.4.2 Designers in AI

From the perspective of the designer, AI is seen as either a tool or as a domain to design for. As discussed in the previous chapter, the designer becomes part of the system and the domain of AI is no different. The question now becomes what the role of the designer is. Design being as abstract and broad as it is, makes this a tough question to answer. It is fair to say that the core design competencies still apply. When placed in a design situation which involves AI, one core competency as listed in the previous chapter changes: working with interdisciplinary perspectives now translates to working with the AI engineers and data scientists who will be creating and implementing the models.

In a conversation with the creator of the AI meets Design Toolkit (see Chapter 4.1), she stated that in her experience being open to learning technical knowledge, and interacting with the engineers is key. Designers should put effort into learning about the other professions and have a broader social perspective. The other core competencies are still abstract enough to not require any translation. However, as discussed in the previous chapter, skills are not generally applicable but are bound to a specific domain. It goes without saying that to be able to competently demonstrate other core design abilities, foundational knowledge of the domain needs to be present to effectively use AI as a design material.

In terms of responsibilities when working with AI, privacy and other ethical concerns are of greater importance compared to other domains which border design. Though designers are further removed from the implementation of AI, they have a great influence over it. Even when a designer might not be the officially designated person to worry about data privacy within a company, they should see the influence they have and take responsibility.

Lastly, there is one general skill which, though strictly speaking not necessary when designing for AI, is a major advantage. Having some coding skills allows designers to create basic ML prototypes and improves communication with colleagues working on the implementation side. Specific programming languages used a lot within the ML field are Python and R.

3.4.3 Interviews with Design Students Studying ML

Several qualitative interviews/conversations with design students were conducted regarding their experiences in courses and projects where design and AI overlap. To start, it is important to note that though several problems are quite prevalent, their presence and intensity can differ greatly. This fact is emphasized even more so when considering the students interviewed here share a common background. The following challenges were deemed the most serious and are ranked in order of relevance.

1. Lack of programming experience

Though it is possible to design for ML without having any coding skills to a certain extent, it won't be possible to reach the same level of expertise. Programming is however difficult and a topic of study on its own.

2. No clear link with design

When learning about the technical aspects of ML, attention has to be given to connecting the material to design. Users have to see the relevance of ML for them and more importantly, how *they* can be relevant within the field of ML.

3. No clear link to reality

In the same vein as the last item, placing the learning material in a realistic context helps to cement the knowledge.

4. Mental image of AI/ML being (too) difficult

This is a problem which can vary wildly from person to person and is usually worsened by not knowing how to code. Some people can be negatively affected in how they approach the topic by this mental image (or they won't even want to familiarize themselves with ML at all).

Differences between designers naturally vary even though points 2 and 3 are more general.

4. IDEATION

Using the foundation of knowledge regarding design, education and artificial intelligence laid in the previous two chapters, the focus now shifts to building on this basis by ideating. The ideation process documented resulted in three concepts of which one was chosen to continue the project with.

4.1 Ideation Foundation

4.1.1 Framing the Tool's Approach — ML into Design or Design into ML?

Although the initial project brief set the project's general direction by clearly stating the target group and even the format of the outcome of the project, there was no clear singular problem to solve. Rather, the initial brief simply desired a general improvement of the status quo. Current tools have their place and value, as we will see later in this chapter, but there is always room for advancements to be made.

Designing for improvements however tends to lead to incremental change which is not the sort of result I had in mind. Consequently, this personal drive for a more innovative approach held back the ideation phase. Finding clearer problems and their roots generally tends to lead to more radical change. I therefore made the decision to not start the project with the analysis of ML design tools but instead looked at design, education, AI/ML and their relation before diving into existing tools. The previous two chapters document this exploration while an analysis of design tools was only done right before the ideation phase. This chapter frames the discoveries of the previous chapters into the ideation phase. This broad analysis of the various related topics yielded several problems to tackle.

First on the list are the difficulties and problems experienced by the primary user group as laid out at the end of Chapter 3. The problems of expecting ML to be too difficult, the trouble of seeing their relevance within ML, the sometimes unclear link with reality and a lack of coding skills were deemed the most influential. Most of these speak for themselves but they can be analyzed more closely. When combining these problems with the expansion of design as described in Chapter 2 the cornerstone of the concept becomes clear. Design (education) has expanded immensely by uncovering new areas of design but also by bleeding into established fields such as AI. On a micro/educational scale, the direction of 'flow' reverses: instead of bringing design into ML, ML is brought into design. Meaning, students start at ML and eventually make their way to design. This distinction between what direction the 'flow' is going might seem illogical, after all, the overlapping area between design and ML will be the same. Seeing designers as a bridge between disciplines when it comes to this project is important. To form a bridge, you can start building from both sides, the question is whether ML should be put into design or whether design should be put into ML.

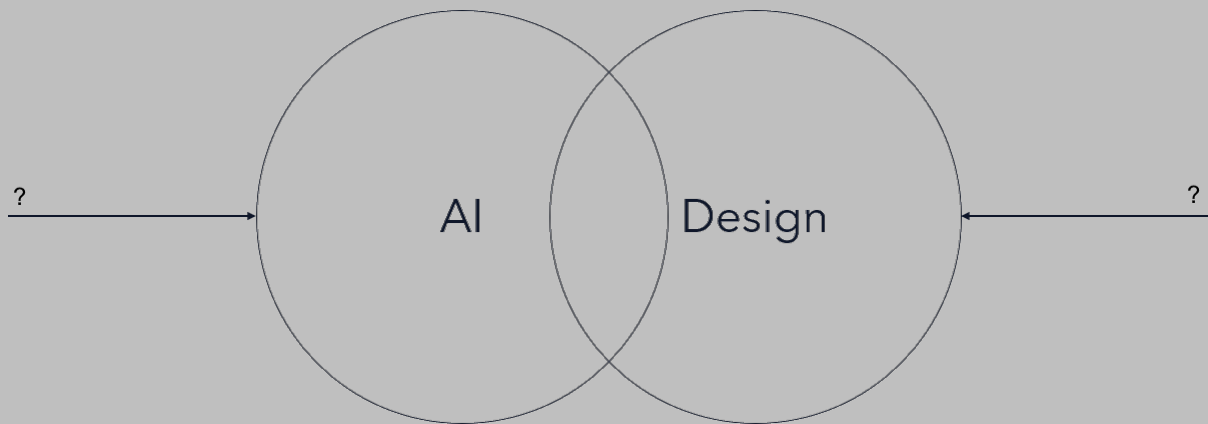


Figure 5: ML into Design or Design into ML? Does it matter if the resulting overlap looks to be the same?

Looking back at the problems, this current approach of ML education in design can explain most of them. It can therefore be hypothesized that reversing the ‘flow’ would help to solve them to an extent. In more concrete terms, this realization solidified the *design first approach* sought during the ideation process. The toolkit should therefore not be too far removed from the domain of design and the starting point needs to take into consideration the foundation designers have and help expand into AI/ML.

4.1.2 Lessons from Other ML Design Tools

After having researched the themes of design, education, AI/ML and their relation, an analysis of several ML design tools guided the project further. When it comes to design toolkits there are several types to be found.

The most common is the ‘deck of cards’ design toolkit. Within the scope of ML, the *IDEO AI Ethics Cards toolset* aims to: “The following exercises help ensure that the work is more ethically responsible, culturally considerate, and humanistic. The activities do not offer definitive direction, but rather initiate a set of actions and prompts to stimulate conversations.” (IDEO, 2019). Already it is clear that these tools are at times prescriptive and at other times descriptive. The topics are varied and pertain to the entire ML design process. Clear downsides when using this particular tool are the amount of text (perhaps the tool should have been a workbook instead) and the lack of any guiding structure. Any topic has the potential to be interesting and relevant but choosing which one to use at any time can be difficult. Overall, I still rate the card set highly.

Another tool is the *OWNML design canvas* (Dorard, n.d.). This canvas poses the designer several open-ended questions on topics related to an ML design process, thereby being fully

descriptive. It is hard to find any faults with this tool, it simply makes the user aware of which aspects to consider during the design process.

The *AI meets Design Toolkit* developed by Nadia Piet (2019) is more prescriptive and aims to “build a bridge between the disciplines of design and the disciplines of machine learning”. It offers users many worksheets, visual exercises and open questions, in a sense combining both of the previous two types. The aim I had for the tool I would create in this project was the exact same. Additionally, before starting the project and having seen this toolkit, the outcome I expected to produce resembled the *AI meets Design Toolkit* quite closely. Upon reviewing the toolkit, I struggled to find any major problems I had with it.

For this reason, I contacted its creator to discuss the creation of her toolkit and the relation of design and ML. When conducting a post-mortem of the toolkit, the content of the toolkit is still deemed to be relevant and useful, however there were several points of improvement to be found. For starters, “the format is kind of boring, it could be more than a PDF”. She furthermore stated that looking back it would be better to make things modular and to add a tool that can direct you within the toolkit. Signposting what people need at a specific moment and not just dumping information would improve the user experience a lot. The greatest point of feedback she gets from users is related to this: the question of whether she could give workshops, talks, lectures, etc. using the toolkit. From this it is clear that the format of this toolkit apparently does not match user needs. Its PDF form constricts. As she put it herself: “People want more interaction and guidance in taking this information to them.” In a way the PDF format makes sense, almost all other design toolkits share this format. It can be easily modified, printed out, distributed and shared. On the other hand, considering the topic this toolkit relates to, i.e., ML, a more hifi format would seem more fitting.

Within the abundance of purely ML tools, all three of the toolkits mentioned above are firmly rooted in design and therefore align with my approach. In my eyes they however do not manage to completely bridge the gap since they stay too close to design, all the while being perfectly useful tools.

4.1.3 Increasing Interactivity Through Design

Following this interview, a key design problem within the project became the question of how to effectively utilize the affordances of a web app (Chapter 2) to stimulate interactivity and guidance within a design tool. The answer to this question also lies in the answer to how to solve the problems design students face when learning ML, namely, the design first approach.

Design is an iterative and reflective process, stimulating interactivity can therefore logically be achieved by staying close to design interactions. Building on top of interactivity, reactivity builds on this even further by not just responding to user actions but more specifically to user decisions.

4.1.4 Instructional Versus Tool-like

There is however a problem with focusing so heavily on interactions. Ability and skill are the end goal but are not the only means of getting there, as a foundation of knowledge is required (though, as Kolb states, this foundation is not the goal in itself). Seeing this spelled out makes it obvious yet education as a whole seems to be moving in a different direction leading to a neglect in knowledge building by focusing too much on the goals instead of how to get there. This development also influenced my initial (and as it later turned out, wrong) ideas on which direction the project should take as I envisioned an educational tool which focused solely on skill development.

A major design problem early on in the project was the trade-off between making an instructional app versus making something which is more tool-like. An instructional tool is less dynamic, by placing content in a pre-made context, every learner more or less follows the same trajectory. A more tool-like approach on the other hand inherently means it is applicable to a multitude of situations and there does not need to be as much of a focus on conveying knowledge if you can expect people to already have this basis. When a tool is widely applicable it can be used more than once, regarding this project it would mean that students could even continue to use it during their personal projects. These two paths both have their advantages and disadvantages, but on the whole, a tool-like setup sparked my interest more. There is however an error in reasoning here. Seeing these two approaches as opposites is a false dichotomy, when conceptualizing there is no reason to fully adhere to one and disregard the other. Rather, the advantages of both should be leveraged without compromising on the focus of the design. The tools dissected earlier in the chapter also do not position themselves as one or the other.

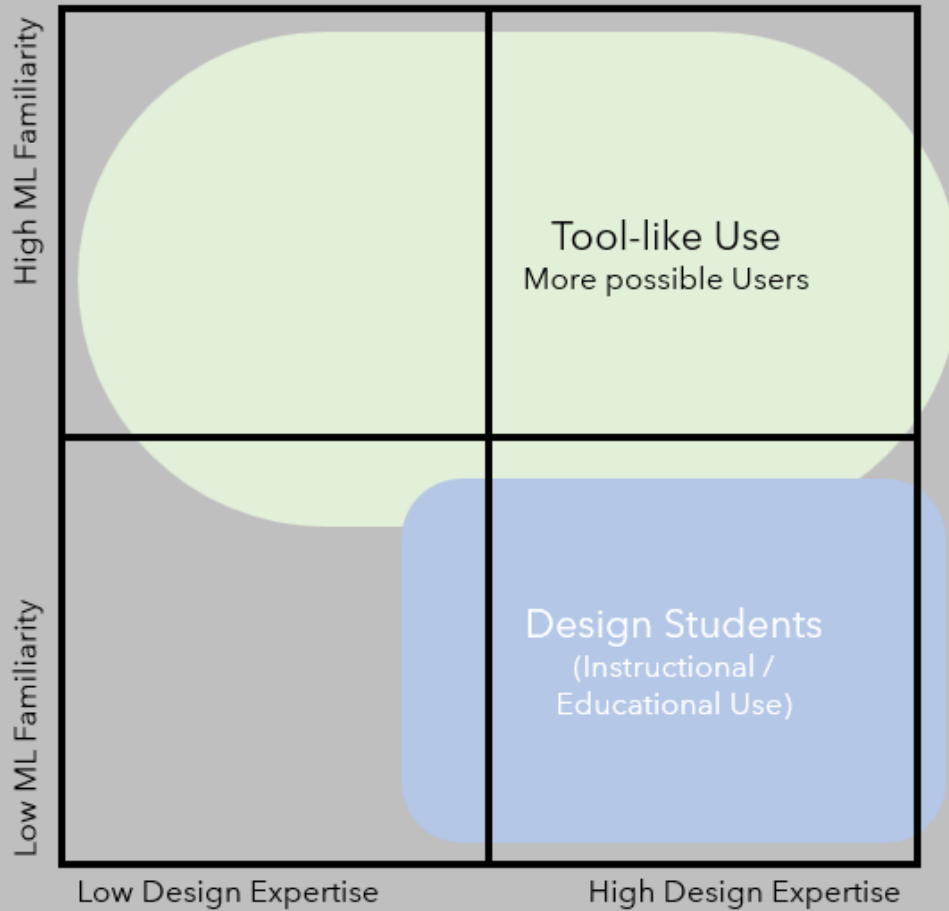


Figure 6: Designing the app to have both instructional and tool-like features furthermore opens up the app to a much wider user base.

These steps should give a rough overview of the process which formed the basis to ideate from further. The initial project brief led a deeper analysis of the current situation to find concrete problems which led to hypothesizing a design first approach is required. Looking at other tools, it became clear that more guidance (for educational tools) and interactivity was a major requirement to improve upon current tools. Luckily, the envisioned digital format of the project's outcome would be well suited to tackle these challenges. Simply aiming for interactivity was however not enough but connecting the interactivity directly to the design first approach would mean that interactivity should be stimulated by staying close to design interactions.

4.2 Ideation Methods

As with any design project, numerous factors contributed to the ideation process. Since the format of the final design is already set, the design's direct context would be the web browser. As such I made templates which were blank, scaled-down versions of web browser windows. These templates helped make the sketches feel slightly more real by seeing the ideas in their possible final context. Since the final design context was already determined at the start of the project, it already made sense to think about how the initial ideas were applied. After sketching many ideas, some were converted into Figma mockups.

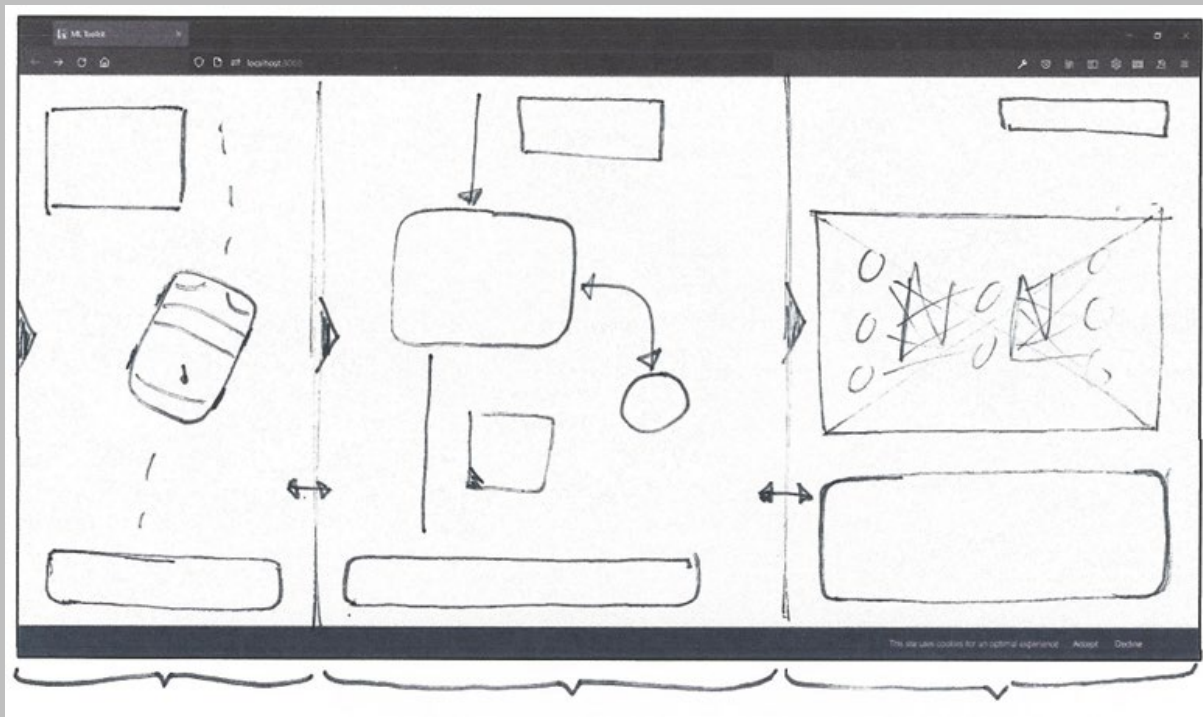


Figure 7: An early sketch of the second concept, ML Design Boundaries, drawn inside a browser window template.

4.3 Concept 1 - ML Design Canvas

The first concept builds upon these guidelines nicely as the first thing the user sees is a provocative question regarding the relation of design and AI, they are given the choice to agree or disagree. A robot coach at the bottom right of the screen reacts to the choice the user makes. The initial idea of the coach was for it to ask the user questions and based on their answers give them tips and insights on how to design for AI.

The left side of the screen is reserved for the canvas and its components such as images, notes, icons, arrows and most importantly widgets.

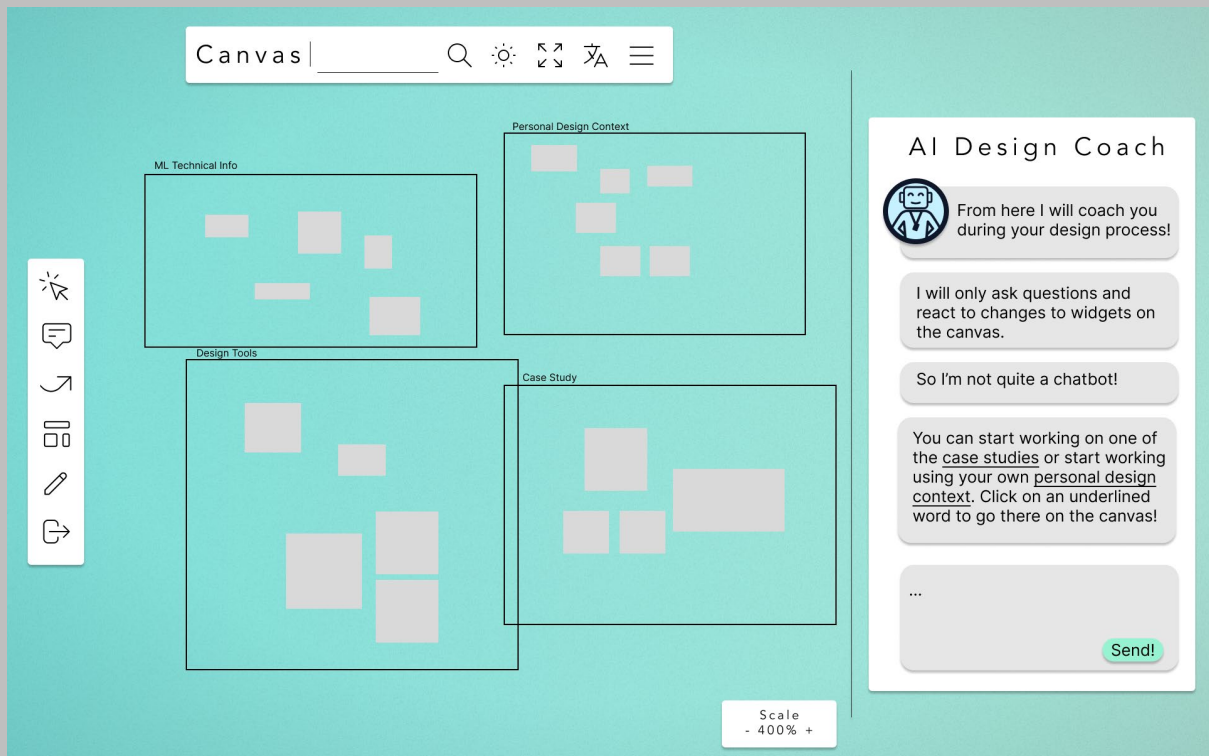


Figure 8: Figma mockup of the ML Design Canvas concept.

Widgets are the interactive and visual representations of design decisions and trade-offs. As for what the specific content of these widgets can be, the toolkits analyzed earlier in this chapter provide useful starting points. The plots, exercises and activities described in the toolkit can be applied well in an interactive design such as this concept. There can for instance be a widget for plotting Confusion Matrices. Users can fill in what true positives, false negatives, etc. look like and what consequences would be. Accuracy percentages from a model running on the webpage can be fed back into and plotted in the matrix. Another widget is the benchmark slider, where users can plot the current human benchmark, industry standard benchmark, minimum confidence levels, targeted benchmarks and more on a scale from 0 to 100. Again, accuracy percentages from a model running on the webpage can be fed back into and plotted onto the scale.

4.4 Concept 2 - ML Design Boundaries

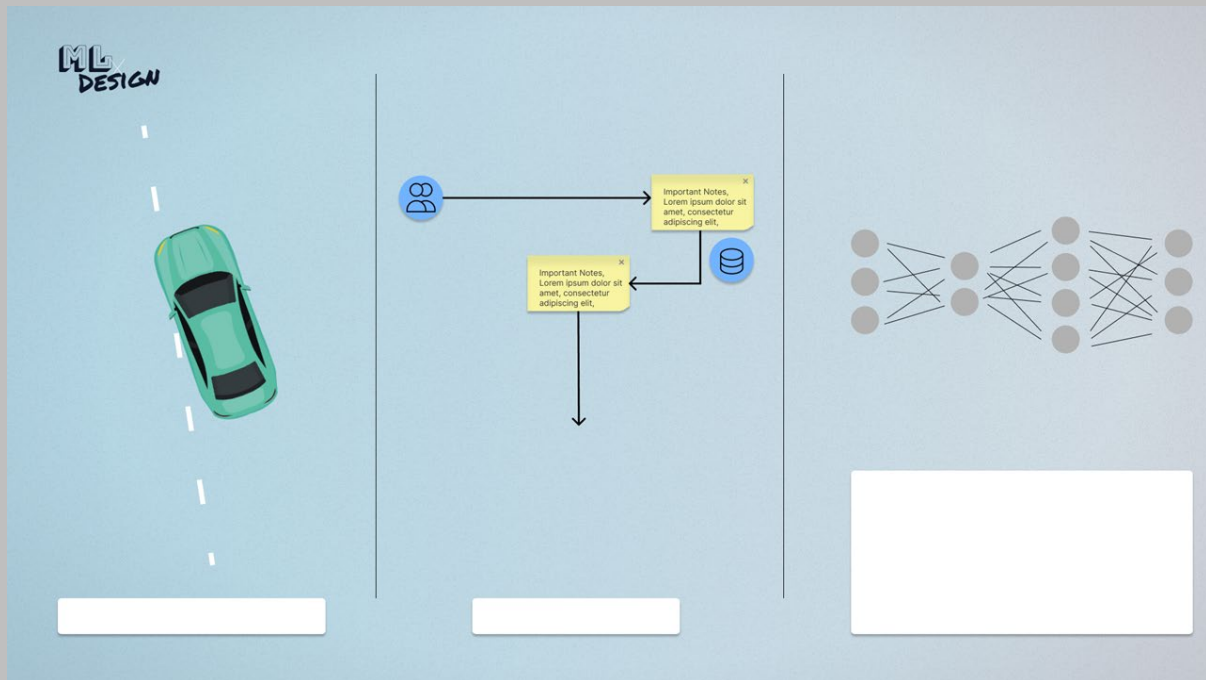


Figure 9: Rough Figma mockup of the layout of the second concept, ML Design Boundaries.

The second concept borrows from the first. An ML design canvas is shown in the middle compartment. This canvas functions much in the same way as the canvas from the first concept but will have limited functionality. Interactive design widgets will be present but data manipulation and the training and running of models will not be required. A neural network is displayed in the right most compartment. The first compartment, the simulator, displays the output of the network, a car which is being guided through a virtual environment by the model. The initial aim of the concept was to teach students how neural networks function. This type of AI was chosen since the architecture of the algorithm is one of the most visual. The concept ended up not revolving around the model however (hence it being positioned in one of the slots towards the side) but rather, the design decisions which influence how the model works take center stage. Based on the feedback from the environment and their own ideas as designers on how the model should function with the help of the design canvas widgets, the network is tweaked. By connecting design decisions in the form of the design canvas between the model and the simulator, the designer is made central to the entire process. Students get to explore what their role is in the AI field when all three domains interact with each other and they can see themselves as the bridge.

4.5 Concept 3 - AI Filter Design

The final concept in turn takes some inspiration from the previous one, revolving around teaching students about the way neural networks function through experiential learning. Neural networks, though still difficult to understand, can be represented visually quite well. The mathematics at the foundation of how the model works can be difficult to understand.

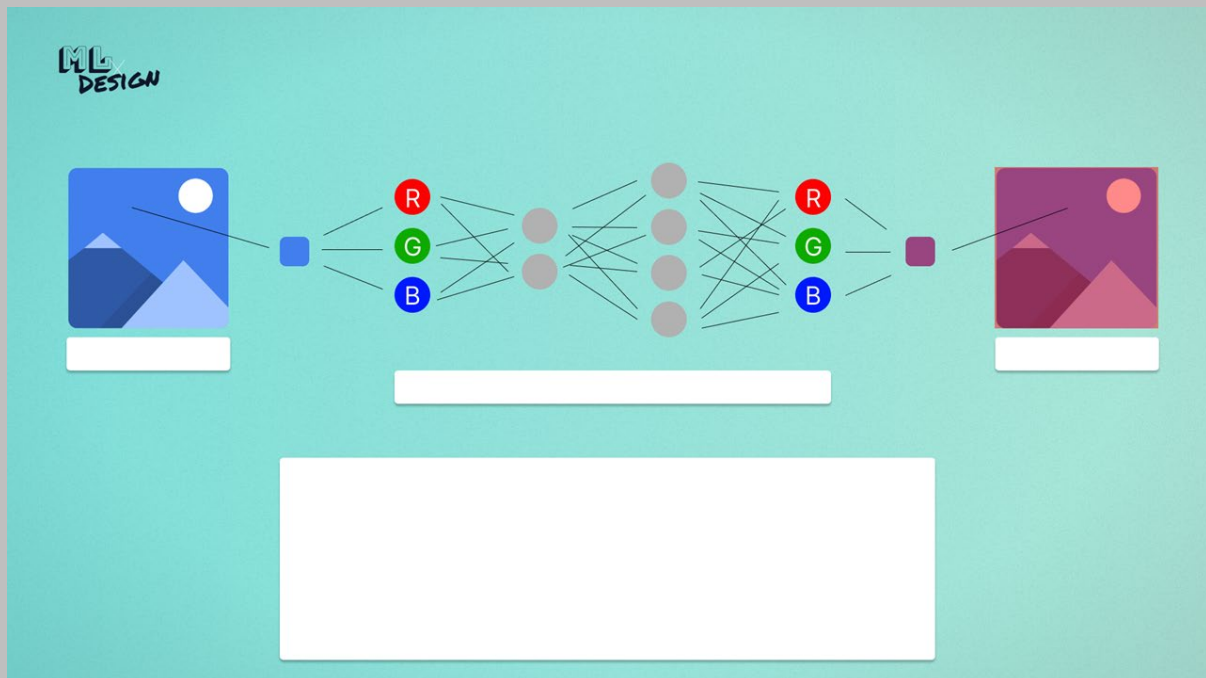


Figure 10: Rough Figma mockup of the general layout of the AI Filter Design concept.

Neurons in a network are not much more than a set of inputs, weights and an activation function. I therefore opted to substitute these 'dry' numbers by colors. Just like inputs are added together, colors can be added together since both are, when it comes down to it, numbers to computers. Students can upload an image or select one from the default ones, and select a pixel in the image. This color is broken down into its R, G, B values which are the model inputs. Then, neurons in the hidden layers mix the colors based on the weights and biases. The outputs are the same components as the inputs and these combine into a color. Though only one input and output color are shown, the way the model operates is immediately applied to all pixels in the output image. To guide students in their learning, the view at the bottom of the screen shows students some interactive questions and sets them several goals. Changing tabs allow students to read up on the theoretical side of the questions and goals they are working on. This concept would not only allow students to understand how neural networks function but also get them some experience in working with this type of AI.

4.6 Concept Choice

After having developed three concepts, the task of choosing which one to continue with was next. All three concepts offered unique ways of working with and learning about AI.

The third concept was dropped quickly even though out of all three concepts I was most excited about prototyping this one. The problem was that I struggled with connecting this concept, albeit a fun and engaging way of working, with design and design decisions. The concept therefore feels more like a fun demo which anyone can use rather than something which is made specifically for designers. It furthermore only applies to neural networks and cannot be applied to other machine learning algorithms, let alone other design problems.

In contrast, the second concept also utilizes a neural network yet it does succeed in connecting design and design decisions with machine learning. Ultimately it fails in being useful in a general setting, in other words, the concept is too narrow and educational.

The reasons for choosing concept number one should however not be seen as the result of crossing out the others. The concept is exciting and should be highly beneficial to anyone in the target group and because of its format it allows for a broader application context.

4.7 Chapter Conclusion

The ideation phase yielded three concepts, diverse yet with some overlap. Of these three, a machine learning design canvas tool was chosen to continue the project with.

This initial concept idea saw students being guided by an AI design helper, interactions with the canvas were secondary and were shaped by said helper. There were several problems with this approach. First of all, the technical implementation of such a feature would be an entire project of its own and might be too complex to develop well enough to be functional given the duration of the project. The name design coach could also bring the wrong expectations and a sub optimally implemented coach would only subtract from the overall experience. The decision to scrap the coach and focus on the canvas instead was therefore easy to make.

The selection of available design for ML tools is tiny compared to the purely ML focused tools. Staying more closely to the design part of designing for ML for the interactive toolkit meant that it would be filling in the gap where interactivity and design meet. Sticking to the target group of designers by not making a generic ML tool was therefore important. This is where

the canvas comes in. Drawing, mapping and brainstorming is supported well by canvas tools. Furthermore, making the toolkit an integral part of the design process was a necessity: the tool needed to cover the entire ML design process/system to offer the desired degree of interaction and reactivity. Using a canvas therefore made perfect sense. There was however one problem with staying close to the design part of designing for ML. Instead of becoming a generic ML tool there was the danger of becoming a generic design tool. Not all affordances of the digital environment are employed at the moment, the most important being computational capability. Future features should therefore be focused on letting users run some basic models within the toolkit.

The design centered approach and perspective of using a canvas would definitely solve some of the problems design students encountered when studying ML such as connecting the material to design and the real world and would be more approachable because of the tool's format.

Coming back to the design coach, in a sense the idea was not completely scrapped. When analyzing what drew me to the idea, a couple of things stood out: unique interaction and most importantly, instant reaction. This reactivity sees the coach respond to the user's decisions to create a feedback loop. When dissecting the affordances web apps might bring digital (design) education, interactivity and reactivity were key. The reactivity was thus added back into the design in the form of the Info View which displays warnings and suggestions based on user actions and decisions but without the hassle of having to create a coach character.

5. **PROTOTYPING**

This chapter covers the prototyping decisions and process which led to the app version as it was at the end of the graduation project. The prototyping phase was the longest in the project and resulted in an app which at this point can be used by students. Future features and additions are covered in Chapter 8.2.

5.1 Determining the Prototyping Goals and Process

Telling fellow students about the concept the responses were positive but these were not actual validation. Creating a prototype would mean that the concept could be validated through actual user tests.

Although the prototyping of what would become the final design took place much later in the project, many of the choices which determined the form of the prototyping process were taken early on and were partly shaped by personal ambitions. The deliverable being a web-based app meant that there were two possible roads to follow for the final deliverable.

Prototyping could be done in a selection of app/UI design suites such as Figma, Adobe XD, Axure or Proto.io, to name a few. These programs allow users to quickly get started designing apps and UI and even offer features to add a degree of interactivity. Though easy to pick up, they lack fine grained control when it comes to adding actual functionality. Their ease of use meant that initial mockups were made in Figma during the ideation. From the beginning of the project, it was however clear that the final deliverable would be an actual web-based application instead of a mockup. A major personal goal was to deliver a design which could immediately be used by students. Other personal ambitions, stated in the project brief, solidified the choice to code the app from the ground up even further: the personal interest in programming and the challenge of learning how to use technologies new to me being the most significant. Moreover, ML being a main topic within the project, a more functional, bespoke made, application would be a better fit. Digital media allow for interactivity, reactivity and computational abilities. Another advantage of choosing this route means that it might give the project more longevity post-graduation as it would be easier to continue working on an app than it would be to take mockups and start from scratch. Looking back, this prediction was correct and the detailed state of the prototype was one of the project's greatest strengths.

In the initial stages of the project the plan was to have created several functional prototypes of different designs around the halfway point. Given that each prototype, though different, would utilize the same tools and technologies it made sense to first build the foundation from which every prototype can be loaded. Creating this web environment in the early stages of the project furthermore would slowly build familiarity with some of the technologies.

5.2 Tech Stack

5.2.1 Web Development Basics

The development of web apps is often divided into two parts, front end and back end. Front end deals with the part of the website the user sees and/or interacts with. Back-end development is concerned with servers and databases and is everything the user doesn't see. Full stack is a term to denote the combination of the front end plus the back end. The list of different development frameworks and technologies used in the complete stack is referred to as the tech stack.

When creating a website, no one starts from scratch. There are thousands of frameworks, packages and libraries which can do much of the heavy lifting by providing code, templates or other useful features when working on larger projects such as debugging, testing and creating a final build to name a few. A major downside to these additional systems is that it is difficult to switch between systems during development. When a project is made using a certain framework, you are locked in. For this reason, carefully choosing a tech stack is important.

5.2.2 Current Tech Stack

The most influential decision in terms of the tech stack is which front-end JavaScript (JS) framework to use. Having discussed the project with a full stack developer, Vue 3 was chosen for this project's front-end JS framework. Vue positions itself as '*An approachable, performant and versatile framework*' (Vue, 2022). This versatility meant that it was possible to make either a single page application (SPA) or a server side rendered application (SSR). Both have their advantages and disadvantages. Seeing as the prototype would likely not require many routes (pages) an SPA approach was chosen. SSR apps however have better initial load times and are excellent for search engine optimization (SEO). A hybrid approach is possible and although this has no priority at this time it could prove to be useful when opening the app up to a wider audience. Learning how to use a front-end JS framework was one of the major personal goals I set for this project and Vue offers a perfect balance between offering an approachable way of working and an active community.

Choosing Vue automatically locked in other development tools. To quickly summarize: Node.js is an open-source runtime able to run JS code outside of a browser. The Node package manager, npm, is a web repository for Node.js projects, and a tool which lets users download and install said open-source packages, Vue being one of them. Here, many plugins for this

framework are also available, one of which, the Vue Router, allows quick and easy navigation (or routing) between pages within the application. Node also handles dependency management using the package.json files. The next tool is Vite. Besides bundling modules and running development servers, this tool handles the creation of the final build of the application so that it can be deployed to a server. Tailwind is a CSS framework which offers quick and customized styling by using utility classes based on a design system instead of working with arbitrary styling values. Interact.js is a drag and drop library, which is a large factor in making the canvas elements interactable.

Although strictly speaking not part of the tech stack, the project's source files are stored in a public repository on GitHub (a link to which can be found in the references) and their platform was also used for version control throughout development. A number of smaller packages and plugins were also used, a complete rundown can be found in the package.json file in the project's GitHub repo.

There are two major libraries which deliver ML functionalities, the first of which is Tensorflow.js. This library lets users run, build and train models while offering fine grained control. The other is ml5. Built on top of TensorFlow, this library is aimed at providing a more approachable experience. During the prototyping phase these packages were explored and used. The final prototype, as discussed in the following chapter, did not end up utilizing these libraries. Future versions of the app might however involve more of the computational side of ML and the experience gained with exploring these libraries will have been useful either way.

For the sake of maintainability and DX, a deployed version of the application would benefit from changing the JS framework to Nuxt and adding Pinia for state management. Using Express as middleware between this new front end and a Supabase database would create a robust tech stack for a modular and maintainable project. The complete future tech stack (and the complete list of future features) can be found in Section 8.2.

5.3 Prototyping Scope

Having established the prototyping means, the scope and goal of the prototype could be worked out further. As the project brief states, one of the project's goals was to deliver a working design which could be used by students and designers. That is not to say the app should be finished, setting such goals would be unrealistic. Rather, the proof of concept should be there, something which could be expanded upon later. Determining what should be included in this minimum viable product (MVP) evolved and became clearer throughout the

prototyping phase, after all, the design also evolved and was iterated upon after the initial ideation sketches. In short, it was deemed best to completely switch focus to the canvas. Secondly, the reactivity which was lost by removing the design coach was replaced by a warning/suggestion system. This all meant that the scope was cut down giving the prototype more focus and reducing complexity. Other design changes are discussed in detail in the following chapter. Choosing which features would and would not make the cut depended on three things: which features had to be present for the user test, what features would allow students to meaningfully use the app and lastly, the remaining time in the project. The user test was done at the end of the project and is described in detail in Chapter 7. The initial prototype scope called for a complete vertical slice (VS), this however turned out to be too lofty a goal. A complete VS would call for a functional back end where users could login and download and/or upload their canvas creations. A server-side implementation of this process turned out not to fit within the project's timeframe. This does not mean the app cannot be used in a meaningful way already, the saving and loading of canvases can be done fully client side (on the user's device without the need for servers) within the application, thereby satisfying one of the major, and most difficult, requirements. Other feature requirements for the prototype were: basic canvas navigation, reusable canvas element object model, ML design widgets, UX (user experience) challenges, image elements, menus from which to insert these elements and the info view window where the user can get extra information on the canvas and its elements. Several additional requirements were made after the demos to experts, these demo sessions will be discussed in greater detail at the end of this chapter.

5.4 Ensuring Longevity

Early on in the prototyping phase it was thought that the app to be deployed will be a direct continuation of the prototype. Consequently, besides having set up a carefully crafted tech stack, some other measures were taken to ensure the project remained modular and maintainable. After conducting more research, it was decided that the first publicly deployed version would be an entirely new app after all. This version will however still aim to deliver on the concepts below.

5.4.1 Maintainability

Building an application which can be used in the long term means that it should be made in such a way that in case other contributors were to join the project they would be able to get a grasp on how the system works, without the help of the original creator. Such a requirement is hard to test but there are a handful of measures which can be taken. The first is documentation, both internal and external. Writing so-called self-documenting code and using

comments are examples of documentation within the codebase and help to make code understandable on a micro scale. Unit testing also belongs to this category by demonstrating the expected behavior of code units. Comments were used in the files of the most important elements such as the canvas and the canvas object template. As for the quality of the code, this is another point which is unfortunately next to impossible to test. External documentation is any documentation that exists outside of the code base. In the case of this project, it means that at the time of writing this report, external documentation is limited to what is found in this report. Lastly, as has already been argued in this chapter, migrating the project to the frameworks in the envisioned future tech stack would benefit maintainability greatly by improving the DX.

5.4.2 Modularity and Expandability

The strength of the design could be said to reside in its mini design tools: the ML design widgets. Such a collection of tools won't be exhaustive. The process of expanding this set of tools was therefore given extra thought. All canvas objects inherit from and slot into a template canvas object (or, in case the canvas object is a widget, a template widget object which in itself slots into the template canvas object). This ensures events and interactions are consistent across all object types. The state of the current prototype is as such that the API which lets objects send warnings or suggestions to the info view is quite tedious to implement and needs significant refactoring. A state management system such as Pinia would improve not just the current system but the creation of new warning or suggestion events within widgets too.

5.5 Prototyping Methods and Phases

Before starting the prototyping phase I sought to find a way how to approach a larger project in regard to software development. There are many paradigms and frameworks to choose from, some being more focused on principles, others being wholly prescriptive. The choice fell on two methodologies on opposite sides of this spectrum. Kanban and Agile development.

5.5.1 Kanban

Kanban is a practical management method which visualizes work and manages tasks efficiently by sorting them in three different categories, To Do, In Progress and Done. The Kanban method has its origins in the automotive industry but has been adopted by the Agile community in recent years. A Kanban board was created at the start of the project on the Trello platform. Reflecting upon the implementation in the project, the conclusion is somewhat

disappointing. The Kanban board was used infrequently which simply defeats its purpose since an incomplete group of tasks fails to provide any overview. Looking back, there were three main reasons for its use being irregular. First of all, the board was too general by covering all tasks related to the project instead of sticking to prototype development. Having an overview of a project is great for communication between team members but as for me, working solo meant that this need was not present and all tasks were kept track of in my head. The most significant reason however for its infrequent use was that the board was not integrated into the development environment. Looking back, it is a shame the Kanban was not used systematically since it would have brought a greater focus to which tasks were truly important. The final issue of a lack of integration was solved at the end of development by downloading an extension for Visual Studio Code created by Lalit Bauskar (extension ID: lbauskar.kanban).

5.5.2 Agile Development

The Manifesto for Agile Software Development (2001) is a short brief containing four values and twelve principles on how to develop software. Compared to Kanban, the Agile Manifesto does not prescribe any processes in specifics, rather, the Agile way of working can be likened to a mentality. The document was studied right before the start of the development of the prototype and somewhat similar conclusions can be drawn when reflecting on its role in development. Concerning the four values Agile presents not all mesh well with the project. The core values of ‘individuals and interactions over processes and tools’ and ‘customer collaboration over contract negotiation’ did not apply as much to the prototyping within this project. The development was done solo and as the creator I chose the features *and* implemented them. As became apparent earlier in the chapter, the value ‘working software over comprehensive documentation’ was followed, as was the last value, ‘responding to change over following a plan’. The prototyping process after all was intended first and foremost to validate the design. Stripping away the human and business centered values which did not apply to the same extent within this project changes the picture. The honest explanation here is perhaps found in the fact that the prototyping process is inherently agile. As with the Kanban method, if the design principles would have been integrated into the development environment more attention might have been paid to them.

Although both methods were not adopted to their full potential, I am determined to use them for future projects, taking in mind the reflections above.

5.5.3 Prototyping Sprints

The prototyping phase was done in multiple sprints, these were feature-based rather than time-based (meaning the emphasis was on the feature not the timeframe in which to get it done). The initial sprint concerned the canvas features such as moving and scaling. From there it followed to create a basic element which could be selected, moved, resized and edited. Next was creating the base widget object and most of the ML widgets. Having made all the basic elements, the menus were created. Saving and loading functionalities were the last features implemented before conducting a set of demos and interviews with experts; these are all discussed in detail in the following section. Listing the sprints like this paints a more structured picture than the process was in reality. First of all, there were multiple instances where significant pieces of the code base had to be refactored to ensure easier development later down the line. Besides this, often multiple features were developed at the same time.

5.6 Demos to ML Experts

5.6.1 Demo Setup

As an initial round of concept validation, the unfinished prototype was presented to several experts (n=9). These were faculty members who have experience working with and teaching design and ML. There were no differences between the materials presented during the demos. Based on the insights gathered from these demos, the features to implement in the final sprints became clear. As such, the state of the prototype during this demo period will not be explained but subtracting the improvements should paint a clear picture. The aim of the demo was to validate whether the tool could be used in both an educational and project setting.

Format wise there were some differences: besides being either online or offline, I found that demoing the concept to multiple experts worked well as they would transition into ideating more easily instead of only providing feedback.

5.6.2 Demo Procedure

Each demo was split into three parts. A short five-minute presentation gave the experts some background on the project. After the presentation a ten-to-fifteen-minute live demo of the functionalities of the prototype was given by importing a canvas made specifically for the demo. Finally, the concept was discussed.

5.6.3 Demo Outcomes

The general reception of the concept was highly positive. All notes taken from the interviews can be found in Appendix B. At the start of the demo, almost all participants were immediately surprised by the level of polish of the application even though it was still an early prototype. Most immediately saw clear value for people beyond the target group of design students and thought it should be presented as such. Compared to design students, the experts who were shown the demos had more varied backgrounds which led to interesting perspectives based on their field of study and line of work. Still, there was a considerable amount of overlap in feedback and their comments are therefore clustered around several central topics and will be presented as such below.

Case Studies and Templates

The demo given to the experts was an overview of the features which were implemented at that point. The premade canvas was not a proper representation of what users would see or create. At the time, there was only this premade canvas and a completely empty one. The experts were told that eventually, the goal was to be able to load in case studies. Already after the first demo discussion quickly turned to what the exact form of premade canvases should and could be and the versatility of the concept meant that there were many options. Canvases could contain:

- A full design brief in combination with the necessary context/background information
- A full design brief, background information and frames containing empty ML widgets with some pointer notes here and there
- A fully completed and filled in case study canvas. New users could look at something like this to get an idea on what the app has to offer and how it could be used to design smart products

Loading in an empty/blank canvas will of course remain an option too. The options above build on each other, starting from a descriptive approach all the way to a complete prescriptive one. These options also fully take care of the design problem of how the app should balance its educational and tool-like sides as described in Chapter 4.

Having established that all options are valid possibilities which should be integrated into the app, the discussion changed to the topic of the following cluster:

Guidance, Flow and Structure

The topic of guidance within the app was likely the greatest point of feedback between all demos. At the time of the demo, canvas items and ML widgets could be inserted from a simple

list in the UI without any other information. One of the discussions highlighted the clear need for a guiding setup menu upon loading the application. Beyond UI problems, the overall program flow was also not immediately clear. To a certain extent, the experts agreed that well-made case studies would provide this flow but deciding when ML widgets come into the picture or where the converging and diverging parts of the design process would fit in was hard to visualize for some. Others expressed the opinion that the ML process should be the core of the program flow, another thought it best to base the user flow on the human centered design cycle.

While all the above comments are valid, participants also agreed that the strength of the concept is in its freedom. Finding the right balance is therefore important. One expert phrased it as 'managing freedom' (which is an interesting paradox), further stating: "Flexibility is good, defend it, see it as a possibility". One interesting idea was to let users plot themselves on a matrix with the axis familiarity with ML vs position in the process. Based on their position the app can give a recommendation on where to start and with which canvas template. Using such an engaging solution, the user gets guidance into which case study might apply to them.

Reactivity

Some experts were impressed with the application but felt its raison d'être could be more pronounced. Comparisons with other canvas tools such as Miro were made by multiple experts since besides the ML widgets, the application did not offer much else to set it apart. According to them, extra reactivity nicely integrated into the flow would "...elevate the concept to the next level, a demonstration would hit it home". Making the tool reactive was part of the initial idea. This feedback turned the idea into a proof-of-concept requirement for the final version of the prototype. As such, the benchmarking widget was made fully reactive. One valuable idea suggested by an attendee was to have UX challenges pop up automatically based on user interactions/decisions, as well as flashing warnings on the mini-map. Another was that one widget could open another relevant one to prompt the user to go to the next/previous step/widget in the design process. Besides improving reactivity this would also aid general app flow immensely.

Naming

Two types of feedback concerning naming were given. First of all, from a branding perspective the concept needed a real name. The Interactive Machine Learning Design Canvas as it was at the time called, or rather described as, was not particularly powerful. The other was from a UX perspective: concepts such as a confusion matrix might be correct and well known to ML practitioners, but to designers using the tool purely from their background it would not mean

anything. That is not to say such terms should be avoided, considering the tool is meant to bridge the gap between design and ML these should absolutely be taught. Similarly, being more descriptive is important, for instance the word 'model' should be changed to 'machine learning model' according to one expert. At any point in the final application the users should have a (clearer) idea what they are about to interact with, from both an ML and design perspective. In the case of the confusion matrix, blending with known design tools about true/false positives/negatives would be worth looking into; this also applies to other ML concepts. This last point of feedback was not acted on during the project but remains valuable for its possible continuation.

Prototyping

In practice, smart products often use multiple models and the canvas concept is ideal for this. One expert in particular placed a heavy emphasis on the importance of realistic ML use through prototyping. "The crux is that for learning ML you need to move to prototyping more". Features geared towards prototyping such as having interactable pre-trained ML models were part of the initial idea but were missing from the prototype due to time constraints. Providing users with multiple pre-trained ML models so they can use them to discover input/output relations was strongly suggested to really set the concept apart.

Data

To properly prototype (mockup) data is a necessity. In half of the discussions after the demos data was a major point of conversation. To start, this cluster can also be connected to the app flow. Two experts said the following: "See it as having two starting points, either you have data or you don't" and "9/10 times data is the most important part"; the availability of data is a major influence on the process and this simple dichotomy should also be kept in mind when redesigning the app's structure and flow.

Missing Features

Current ML design toolkits use the deck of cards approach, this tool form is not included in the prototype but the canvas format allows these to be easily implemented and would already elevate the format to a next level due to its interactive nature. Some participants requested a frame component to group items as found in other canvas tools such as Miro. In the same vein, making connections between canvas items more explicit would create more order, visualize (data) connections.

(Simultaneous) collaboration is also missing to a degree, this is where the prototype falls short compared to generic canvas tools. Making sure that the app can run well on a big screen with

multiple designers standing around it would allow it to be used in a lot of professional design settings. Touch controls are therefore practically a must but were not implemented in the prototype due to time constraints.

For the slider widgets the trade-off could be made more explicit by scaling the size of the label based on the position of the slider on the track, making one word shrink and the other grow would drive the point home even further that you have to make a trade-off. Besides the font size, the color and weight could also change. Many small changes were suggested when after most demos the discussion of the concept turned into ideation. However, they are left out of this report as they did not relate to concept validation but were UX improvements.

5.6.4 Changes Made Following the Demo

The feedback from the demos was highly valuable. Because of time constraints, not all points of feedback could be addressed. Some of the items listed below were already meant to be addressed.

- The ML Widget menu was greatly improved for better guidance
- The Quick Setup menu was created
- The Info View functionality was expanded
- The Benchmarking widget was updated to be reactive as a proof of concept

These features are explained in more detail in the following chapter.

6. FINAL DESIGN

This chapter details the final design of ALLIXR. For convenience's sake, unless explicitly stated, the final design shown in this chapter is equal to the state of the prototype of the toolkit itself as it was after the improvements made based on the feedback from the expert demos. In reality, there are of course many small points of improvements or features which are not quite implemented. It is valid to say the state of the design is final since the application can already be meaningfully used by design students, demonstrated by the outcomes of the users tests in Chapter 7. Naturally, there are many improvements to be made and Chapter 8 contains a roadmap detailing these and the project's future.

6.1 AILIXR Overview

6.1.1 AILIXR Overview

The AILIXR is a digital canvas design tool which allows designers to explore and adopt the affordances of AI/ML into their own designs, bridging the gap between the two disciplines. Using the many highly engaging, interactive and responsive miniature design tools designers can design smart products, systems and services. AILIXR can be used as an educational tool by loading in case studies or it can be used for real life projects. The design furthermore integrates well into the design workflow and enables designers to communicate their designs to others in their team.

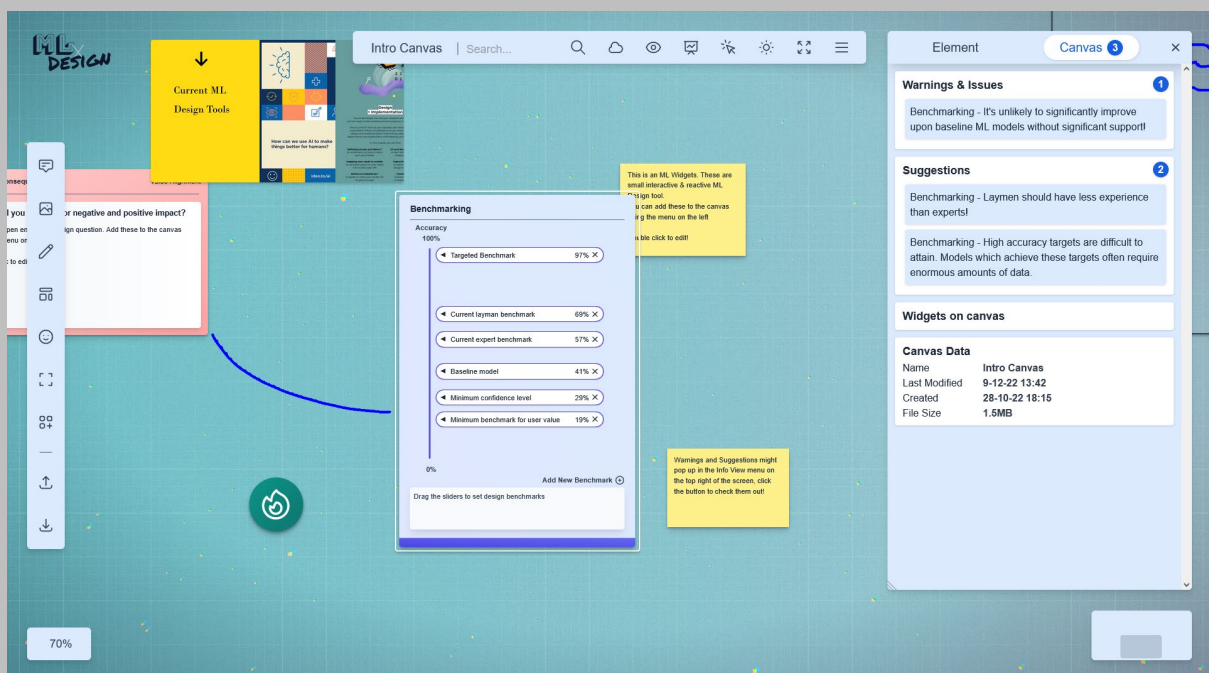


Figure 11: The AILIXR digital ML Design Canvas.

6.1.2 User Interface

Logo

The toolkit's user interface consists of six components. Starting at the top left, the AILIXR logo provides branding and when clicked on directs users to the home page. This branding is useful since others could find the toolkit only having seen a screenshot or video.

Scale Display and Mini-map

The following two user interface components are common in canvas tools: the bottom left card shows the percentage the user is zoomed in (as an aside, whenever the scale is close to its starting scale it snaps back to 100% to improve user experience (UX)). The bottom is a mini-map right card that displays the position and size of the viewport in relation to the canvas. At the moment it cannot be interacted with but moving the viewport by clicking on the map is a possible future feature, as is displaying canvas elements and warnings using the minimap.

Settings Toolbar

The Settings Toolbar at the top concerns general canvas settings. Starting from the left, the current layout is as follows: the canvas title is displayed and is editable. The search field to the right is not functional at this moment but would allow users to search the canvas items and the app's menus. This field is followed by several cosmetic toggles: background visibility, grid visibility, presentation mode, cursor highlighting (also helpful when presenting), a dark/light theme switcher and a fullscreen toggle. Presentation mode hides the UI and activates fullscreen. The last item in the toolbar is a pop-up menu which displays links to general information about the tool.

Interaction Toolbar

The Interaction Toolbar on the left side of the screen groups all actions which have to do with the canvas items themselves. Using this menu, in order from top to bottom, the following can be inserted: notes, images, ML widgets, UX challenges, frames and icons. Clicking the pencil icon toggles draw mode. The bottom two icons are for importing and exporting canvases. Besides importing local JSON files, the Quick Setup pop-up menu displaying template canvases can be opened or a new canvas can be created. Exporting can be done either as a reloadable JSON file or as a PNG image. Images allow for design ideas to be shared more

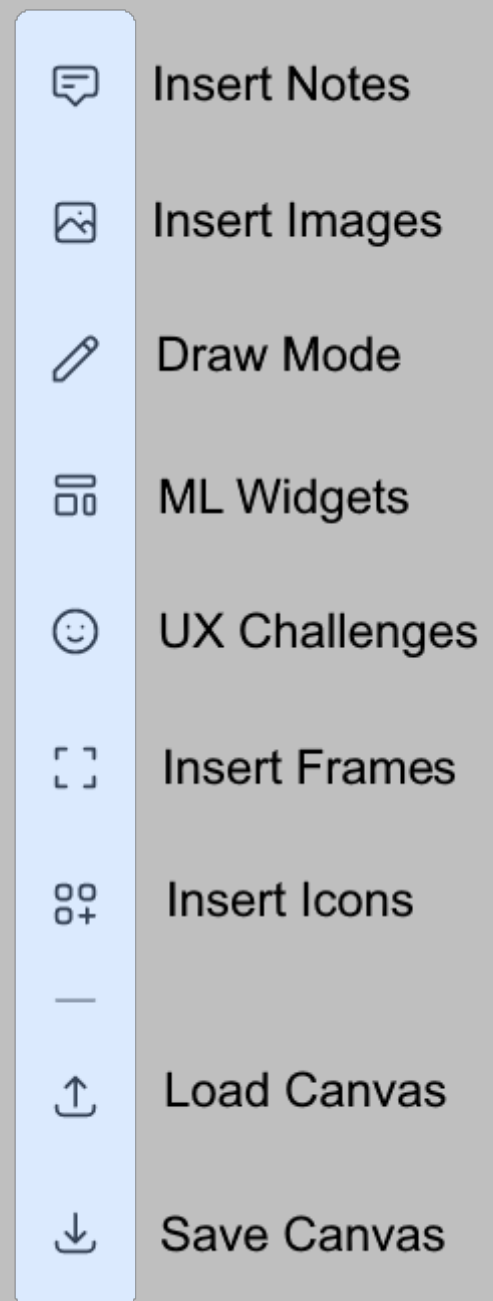


Figure 12: The Interaction Toolbar.

easily, thereby supporting the communicative abilities of the tool. Screenshotting is less optimal since the viewport only covers a fraction of the canvas.

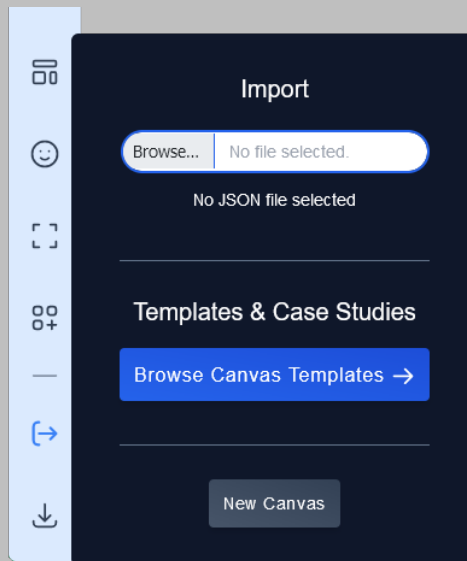


Figure 13: Loading/Importing a canvas using sub menu from the Interaction Toolbar.

Quick Setup Menu

The Quick Setup modal menu is also visible on startup to help users to get started. From here, several templates and case studies can be loaded. The Intro Canvas helps new users learn the basics.

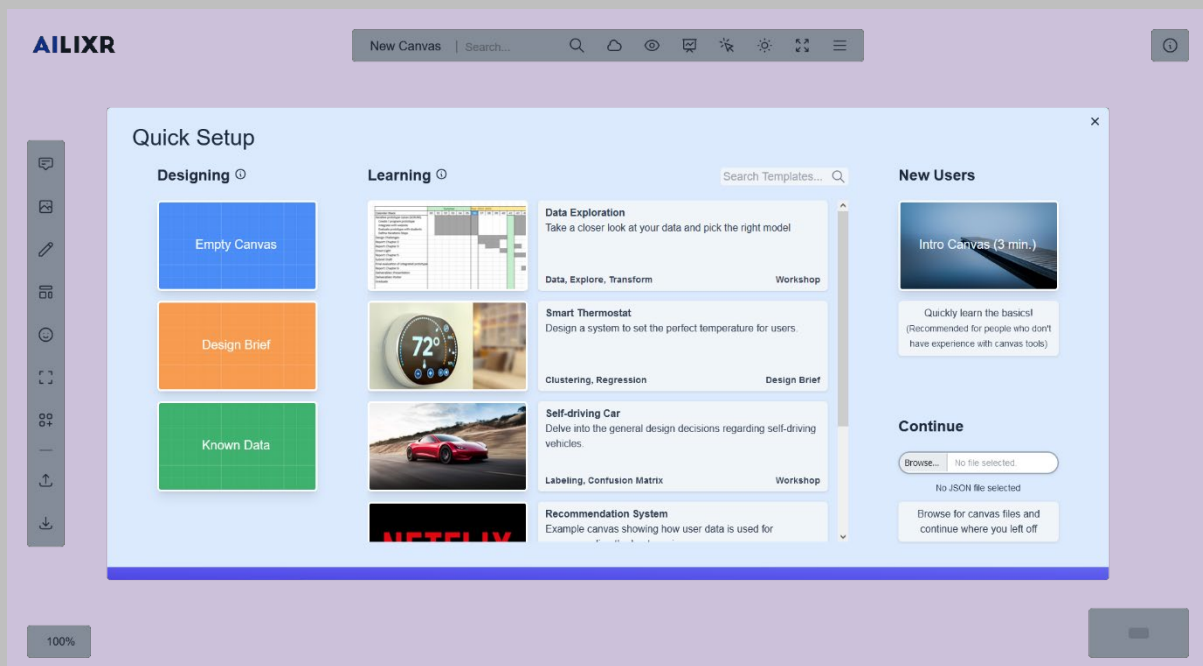
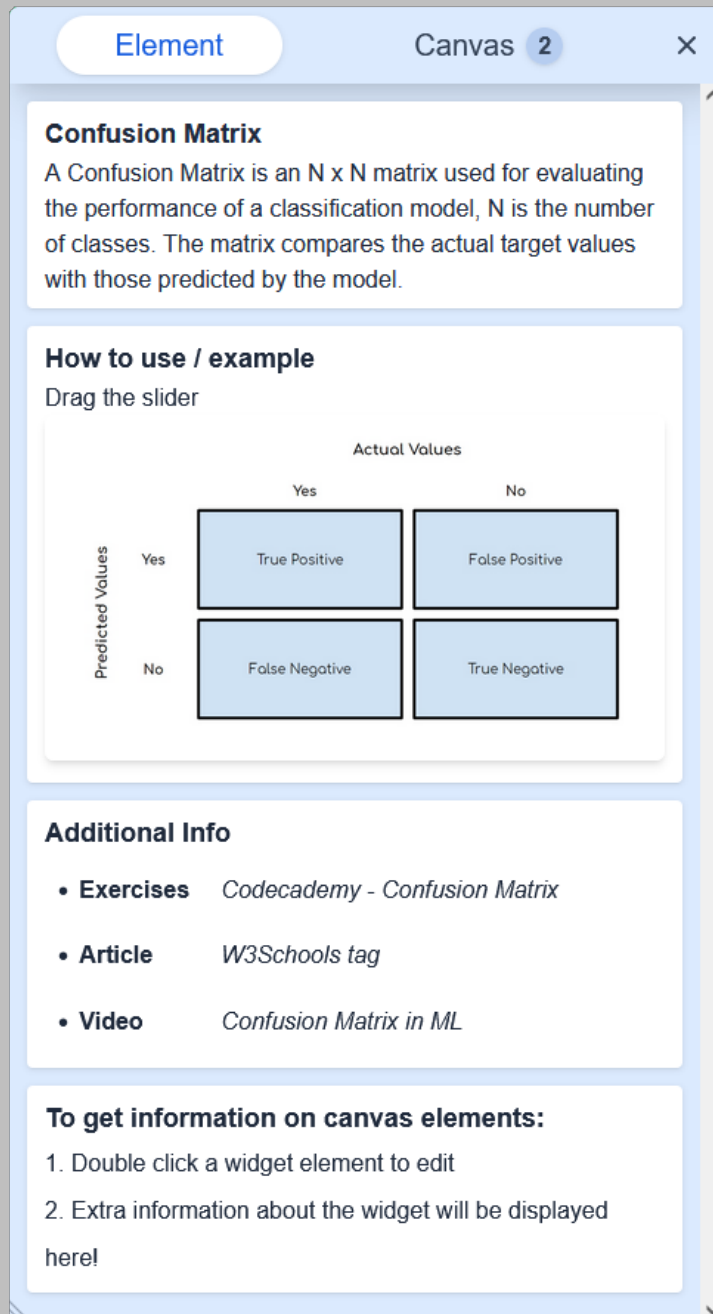


Figure 14: In the prototype, the Quick Setup menu is automatically displayed as a modal when the user launches the app.

Info View Menu

Finally, the visibility of the Info View card on the top right can be toggled. Within the menu, the Element tab displays cards showing a short description, use cases and links to additional information based on the last edited ML widget. The Canvas tab displays warnings, issues and suggestions the user gets based on their interactions with the ML widgets. The bottom card in the Canvas tab shows general canvas info and metadata.



The screenshot shows a window titled 'Element' with a 'Canvas 2' indicator and a close button. The main content is divided into four sections:

- Confusion Matrix**: A text block explaining that a Confusion Matrix is an $N \times N$ matrix used for evaluating classification model performance, where N is the number of classes. It compares actual target values with predicted values.
- How to use / example**: A section titled 'Drag the slider' containing a 2x2 matrix diagram. The columns are labeled 'Actual Values' (Yes, No) and the rows are labeled 'Predicted Values' (Yes, No). The cells contain: True Positive (Yes/Yes), False Positive (No/Yes), False Negative (Yes/No), and True Negative (No/No).
- Additional Info**: A list of links categorized by type: Exercises (Codecademy - Confusion Matrix), Article (W3Schools tag), and Video (Confusion Matrix in ML).
- To get information on canvas elements:**: A numbered list: 1. Double click a widget element to edit; 2. Extra information about the widget will be displayed here!

Figure 15: The Element View inside the Info View menu. This tab displays general information about the ML widget for the user

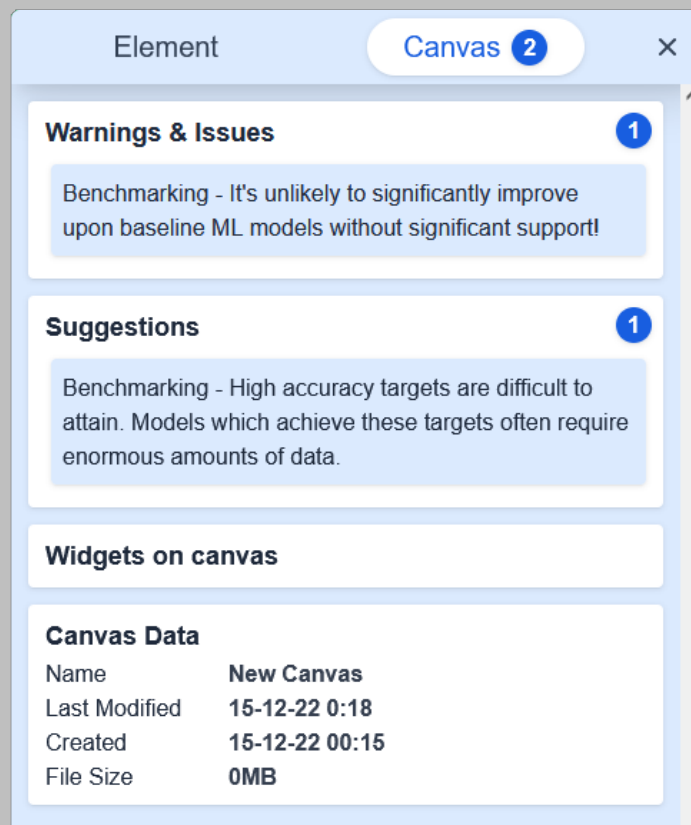


Figure 16: The Canvas View tab inside the Info View menu display general information about the canvas. Most importantly, it communicates warnings, issues and suggestions to the user. Within the prototype, this menu is the only major source of ML reactivity.

6.1.3 General Controls

To new users, controlling the tool should quickly feel intuitive. Even though the control scheme was not copied from any major canvas tools, the highly interactive nature of a canvas app makes finding out how the basic interactions work quite straightforward. Zooming in can be done using the scroll wheel and moving the viewport is done by dragging the pointer with the right mouse button. To make viewport movement feel natural this action is given some momentum. By quickly dragging and then letting go the canvas will continue moving for a moment, a little tweak which improved the UX a lot.

Interacting with canvas items is clear and consistent. All items can be selected by left-clicking an unselected item and selected items can be moved or resized by dragging. Double-clicking an item toggles the edit state so that the content inside the item can be interacted with, regardless of its previous state. Hitting the Escape key moves the state one interaction level lower. Many common keyboard shortcuts have been implemented to increase the UX: hitting CTRL + A / C / V / D for selecting all, copying, pasting and duplicating respectively. Multiple items can be selected at the same time by holding down CTRL when selecting. Selected items

can also be moved with the arrow keys. Finally, hitting Delete will remove any selected items from the canvas.

At the moment only mouse and keyboard controls have been implemented but adding touch screen support is a planned future feature. Touch controls would not only enable mobile devices and tablets to use the tool but smart boards too. Smart boards could provide interesting use cases since they lend themselves well for collaborative design sessions.

6.1.4 Complete Website Overview

During prototype development the focus naturally was on the tool itself. The final design will naturally have a landing page from which users can access the tool. This way the purpose of the tool can be conveyed without overwhelming the user. Login functionality is a planned future feature for easy saving and loading canvases from the cloud. However, access to the tool should not require an account since from a technical standpoint the toolkit can be completely client side (not requiring servers) and requiring an account might deter some potential users.

The rest of the site follows the same basic structure seen everywhere. The creation and design of the site containing the toolkit should not be taken too lightly since external users, namely anyone not using the tool in relation to an AI/ML design course, only have the site to go off of.

6.2 General Canvas Items

As previously stated, (almost) all canvas items can be interacted with in the same consistent manner. This is because they share the same base canvas template item component. The canvas item is then slotted into this template component. Canvas items can be inserted onto the canvas using the Interaction Toolbar on the left side of the screen.

6.2.1 Notes

The first canvas item is the Note since easily adding text to a canvas is a must. The note is styled to look like a sticky note to fit with the design theme. But there is a little more to the note item: if HTML is written into the text area, the markup is automatically injected into the item, styles included. This versatility makes it easy to add custom styled components.

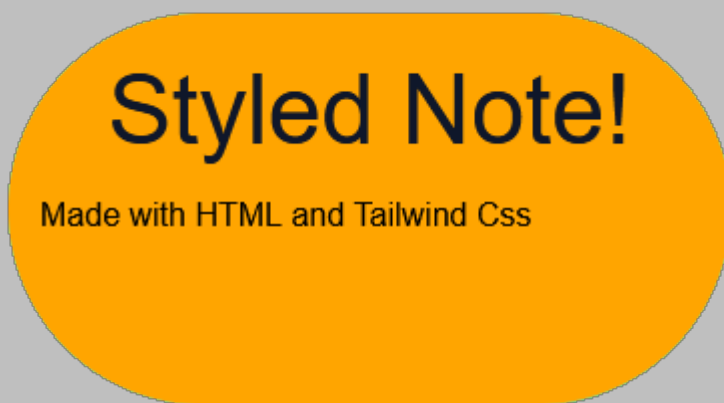


Figure 17: Styled note. Any HTML written on a normal note is automatically formatted.

6.2.2 Images

Images can be added to provide more visual context to projects. These can be added from the user's computer by either using the hover menu in the Interaction Toolbar or by simply dragging the file from their file explorer onto the canvas. Copy-pasting an image from the clipboard unfortunately does not work at this moment.

6.2.3 UX Challenges

The ML Widgets discussed in the following section are interactive tools to specific design questions. This interaction is great but it does not mean open-ended challenges can be avoided. The nine UX Challenges found in the Interaction Toolbar, taken from the AI meets Design Toolkit (Piet, 2019), are great and valuable examples of important challenges which cannot easily be cast into a miniature design tool. Users can also insert an empty template to set and solve challenges pertaining to their project.

How will we help our user understand certain outcomes?

The tag which is created by the model will be given to photos belonging to the user, clicking on it should direct them to a page with informati

Figure 18: Example of a UX Challenge card. Not everything can be turned into an interactive design widget and some open questions are too important to ignore.

6.2.4 Frames

Frames differ from the other canvas items on this list since they cannot be selected or edited though they can be moved and resized. Their purpose is to group other canvas items together to create order. Frames can furthermore be (re)named and deleted.

6.2.5 Icons

The final canvas item type is the icon. The canvas is highly visual, to get the most out of this medium, users had to be given an easy option to add visual items besides images. A small collection of free SVG icons taken from the Heroicons Vue library (Tailwind Labs, 2022) is available to the user from the hover menu in the Interaction Toolbar on the left. This group of icons was chosen because of their general applicability and relevance to ML.

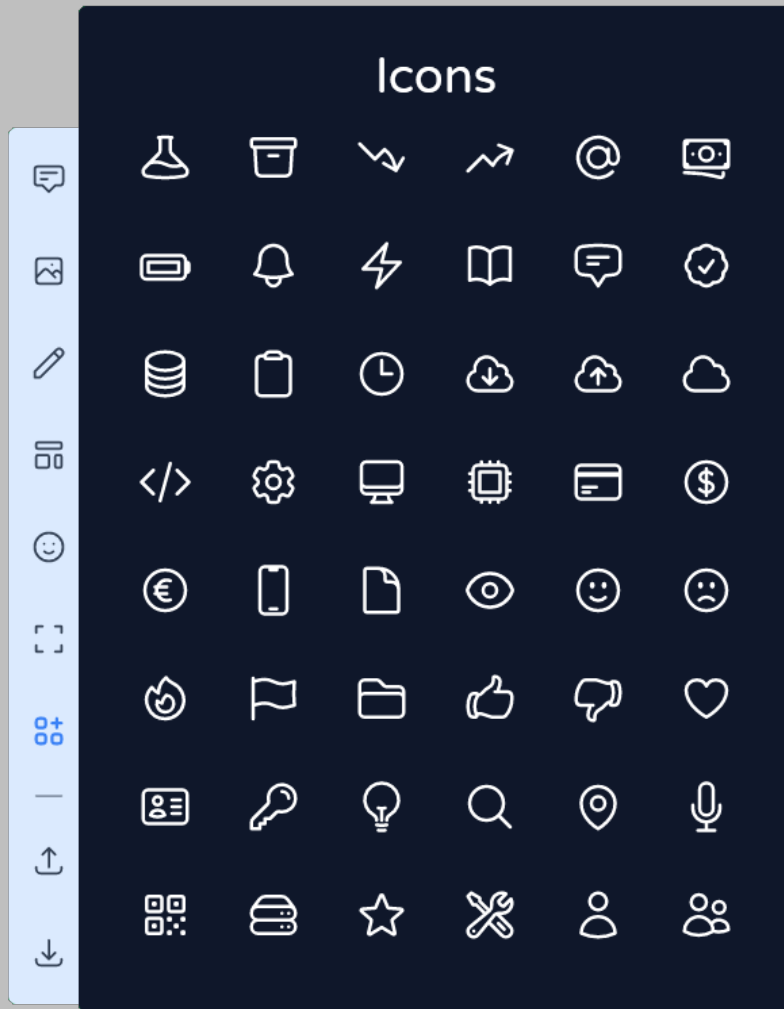


Figure 19: The icons users have at their disposal in the prototype.

6.3 ML Widgets

The widgets are the interactive miniature design tools which turn an otherwise generic canvas design tool into an AI and ML focused design tool. As they are canvas items, they too can be found in the Interaction Toolbar on the left side of the screen. Within the ML Widget pop-up menu, the widgets are divided into three different categories corresponding to the stage of a project they are relevant for. The categories are: Idea Selection, Concept Development and Design Implementation. Hovering over a widget button will display its description on the left side. The widgets as explained below are instead grouped based on their similarities to each other and the type of user interaction. The widget pop-up menu was loosely inspired by the Machine Learning Canvas created by Louis Dorard (2022). Some compartments don't have any widgets belonging to them yet.

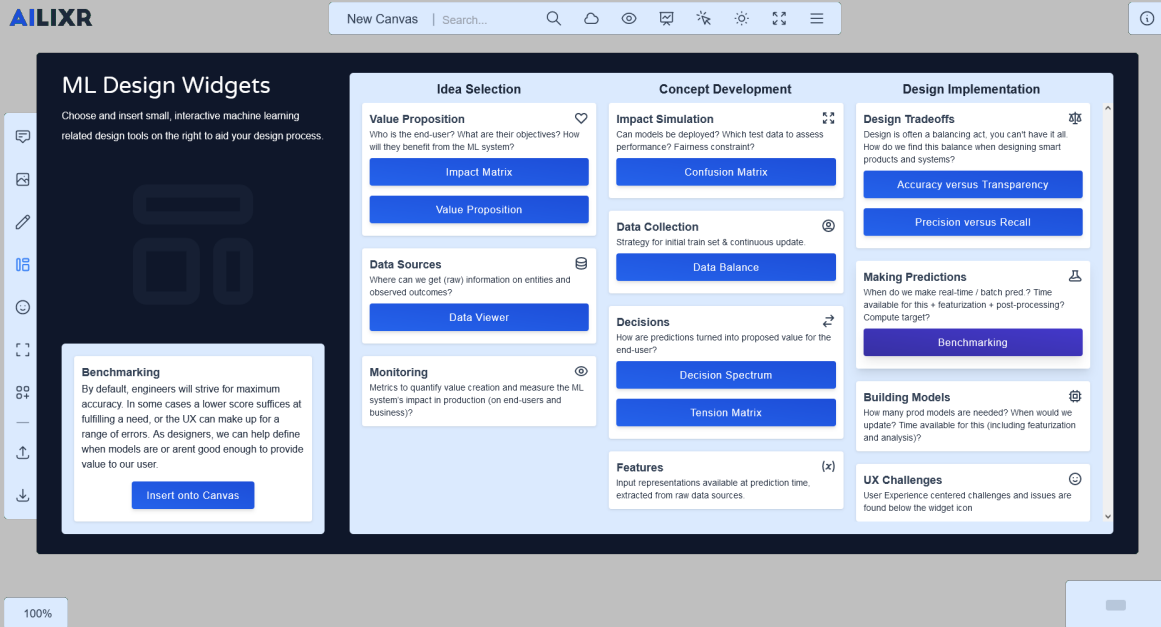


Figure 20: The ML Design Widget menu shows all the available widgets. Hovering over one will show more info on the left side of the menu. Widgets are sorted in three columns.

6.3.1 Templates and Matrices

The widgets in this first category of widgets are grouped together because out of all widgets they have the lowest levels of interaction and are the most straightforward. The last two widgets are not even strictly specific to ML design, nevertheless, when used in a ML design context their value is apparent. After all, the general idea behind the tool was to build towards ML starting from a design foundation. For now, most were taken from the aforementioned AI meets Design Toolkit without major changes.

Value Proposition

Value proposition template cards can be found in many different contexts. As such, the questions the widget asks the user to fill in are generic design factors. The placeholders on the right do relate to ML however. Filling in a value proposition helps to center the focus of the project and allows teams to quickly set design goals.

Value Proposition

Using	data
To	AI capability
We can help	user persona
With a better way to	job to be done
With/without	gain/pain
Because/so that	value

Figure 21: A Value Proposition template is used for many designs. This one has been modified for ML design and is from the AI meets Design Toolkit.

Tension Matrix

This widget is the most straightforward out of all in this section. Users can type opposing values into the labels and can map negatives and positives within each quadrant using canvas notes.

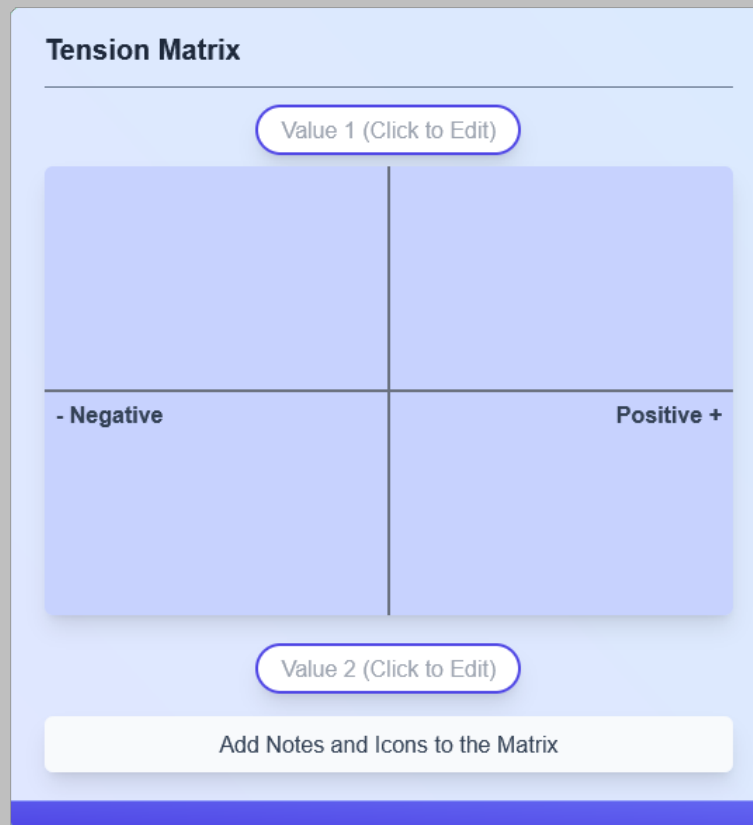


Figure 22: The Tension Matrix is a canvas for plotting polarizing values and what either extreme means for the design. This widget is taken from the AI meets Design Toolkit.

Impact Matrix

This widget concerns the early stages of the ML design process. This widget sees users plot their ideas onto a matrix where user value is plotted against ML investment. This simple tool helps users to quickly compare ideas and concepts so they can choose what to continue with. Horizon 1 is an automatic green light, if concepts fit in the second group more research is warranted. Horizon 3 is to be avoided.

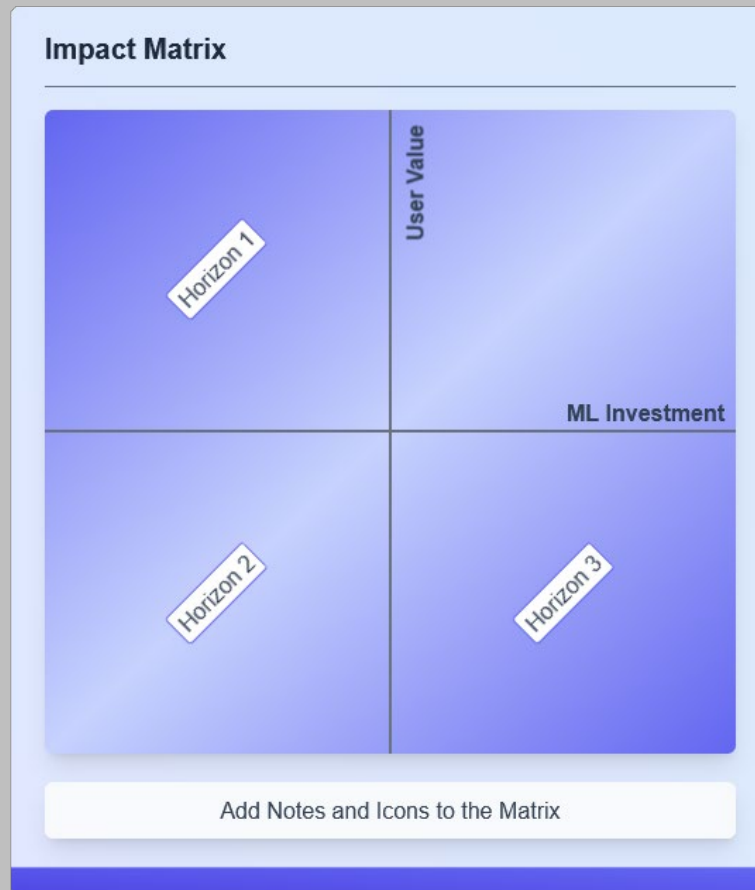


Figure 23: The Impact Matrix lets users plot their ideas to assess viability. This non-reactive widget taken directly from the AI meets Design Toolkit.

Confusion Matrix

Confusion Matrices are used a lot in ML. Normally, this table visualizes the exact performance of an algorithm by displaying true positive (TP), false negative (FN), etc. scores. Such an implementation is a planned feature, the current implementation is geared towards the design implications connected to each quadrant. The designer is brought to consider what false positives look like and what these mean for the design. As with the previous two matrices users can resize the widget to place notes inside but typing is also an option.

		Machine Prediction	
		Positive	Negative
User Reality	Positive	Model correctly says female patient is pregant	Model says female patient is not pregant even though she is
	Negative	Model says male patient is pregant	What does a true negative look like?

Figure 24: The Confusion Matrix widget, this interpretation forces users to think about what each outcome means for the design context. After all, the model will certainly not be completely accurate.

6.3.2 Trade-off Sliders

When placing two items at either end of a slider, increasing the first automatically decreases the other. This simple fact makes sliders ideal as a design tool since design can be said to be all about finding balance; no configuration ever satisfies all aspects completely. Sliders furthermore let users make design decisions which are not completely black or white, positioning the slider somewhere is a preference rather than a concrete target. By focusing on the desired situation instead of a number, the user experience is placed at the center. The current design has two different common ML/design related trade-offs. These are 'Accuracy versus Transparency' and 'Precision versus Recall'. These were both adopted into an interactive form from the AI meets Design toolkit. The 'Decision Spectrum' was inspired by the IDEO AI Ethics Cards toolkit (2019).

Precision versus Recall

Precision versus Recall is an optimization problem. Both can be valuable but the design context will determine which is more important and to which degree. Precision is the percentage of correctly classified predictions (true positives) from the total predictions (true positives plus false positives). Recall is the number of true positives divided by the true positives and false negatives. When it is impermissible to miss positive instances, the model should be optimized for recall, however, when confidence in true positives is desired, the model should be optimized for precision.

Accuracy versus Transparency

The trade-off between accuracy and transparency also exists in almost all ML models. Designers have to ask themselves what is more important to the user and the system: to understand why and how a prediction was made or the accuracy of the model (the percentage of correct predictions)?

Decision Spectrum

The final slider type widget is less of a trade-off but is a spectrum. The difference compared to the others is that neither end has intrinsic value. This slider, the 'Decision Spectrum' widget simply asks users to think about where the decision-making lies at that particular point in the total interaction, with the system, or with the user. This tool can therefore be used multiple times throughout the user journey.

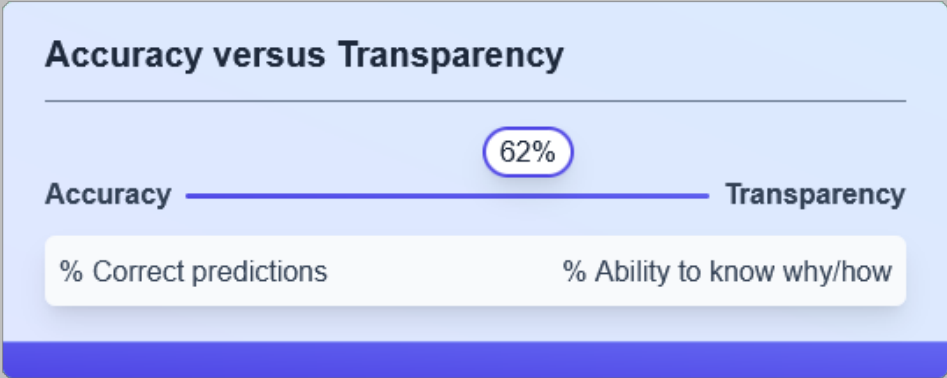
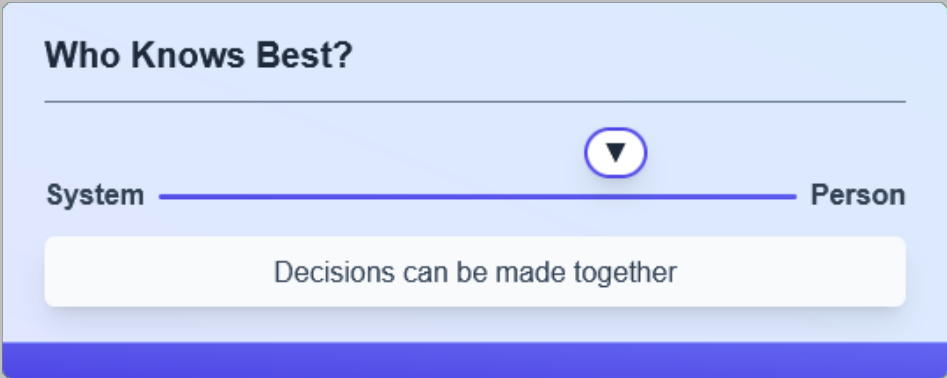
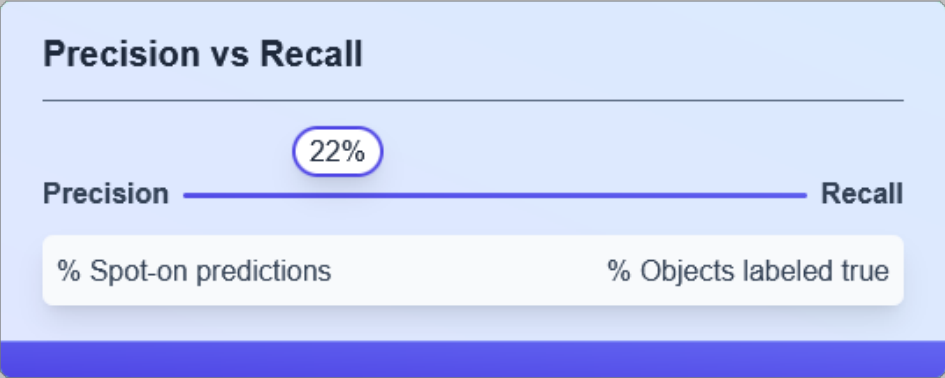


Figure 25: The three slider widgets force users to think about the certain trade-offs they will likely have to make in their ML designs. From the demos it was suggested to scale and recolor the words/concepts on both ends of the range when the slider is moved.

6.3.3 Benchmarking

This widget was given its own section because it is the only widget which for now is developed to be fully reactive. Placing this widget with the other slider widgets was furthermore not fitting since this widget only has a range with one label instead of two opposing sides/values. The widget was also taken from the AI Meets Design Toolkit. The user interacts with this widget using multiple sliders which contain the following accuracy markers: Current expert benchmark, Current layman benchmark, Targeted Benchmark, Baseline model, Minimum confidence level and Minimum benchmark for user value. Users can edit these labels and insert or remove markers.

Its reactivity however sets this component apart from the others: when the users sets questionable or downright impossible targets warnings and suggestions are sent to the info view and a notification appears. Each message is placed in either the warnings or suggestions category and a message containing the name of the widget is displayed. Clicking on a message moves the viewport to the widget which sent the message so the user can resolve the issue. Giving the user immediate and specific feedback on their actions and decisions makes the tool go from being 'only' interactive to reactive.

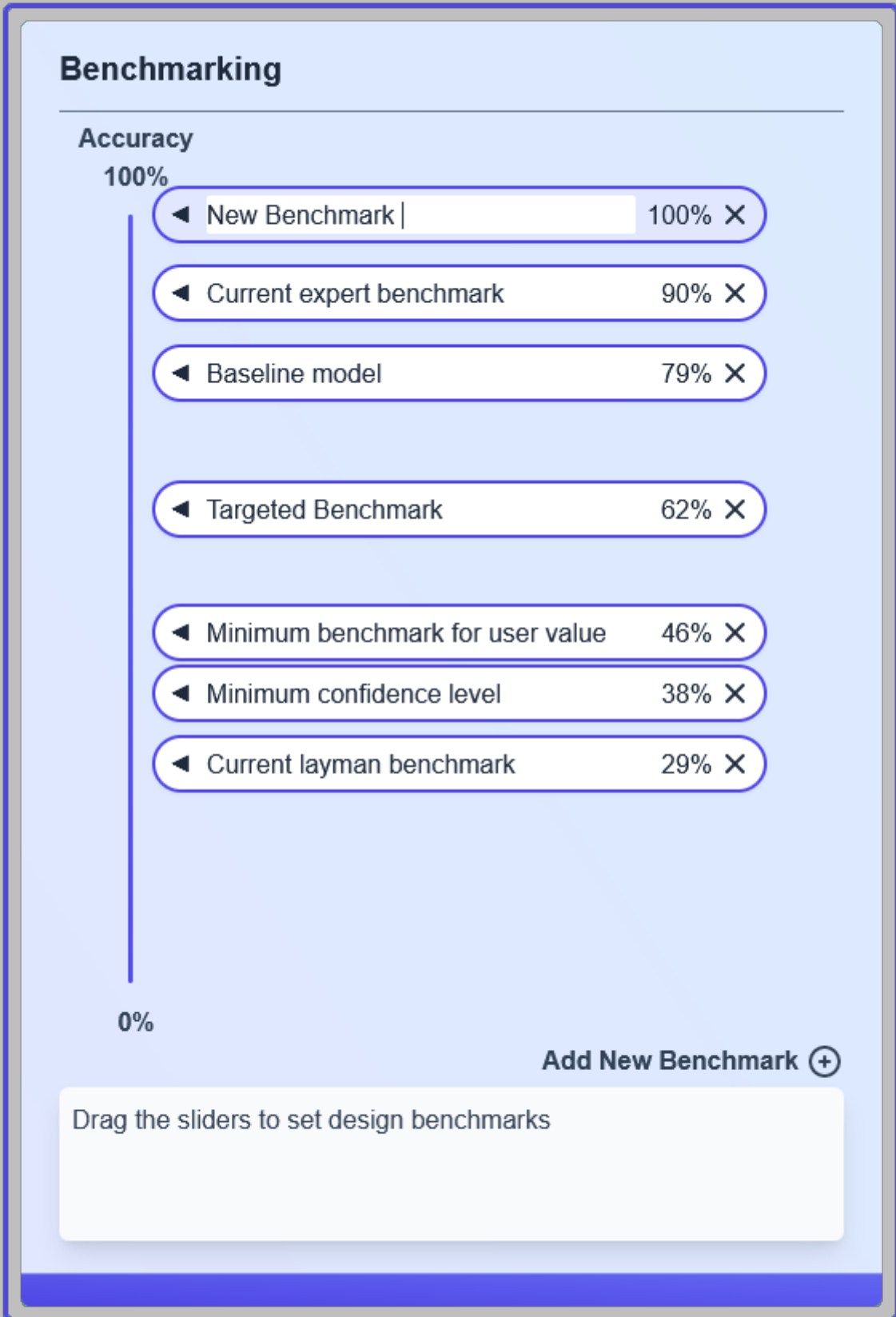


Figure 26: The Benchmarking widget asks users to set benchmarks.

6.3.5 Data Oriented Widgets

Regarding data, users have two widgets at their disposal. Having data is the main ingredient for setting up an ML system after all.

Info Balance Sheet

The Info Balance Sheet enables users to organize their data. Each entry has a description of the data, its source, and the benefit for users. Often, when working with smart systems, data is simply collected without a clear user benefit or need to satisfy. Looking critically at data and tying it back to user benefits can prevent privacy problems. When using the widget users can add or remove entries and move the slider. This slider splits the list into data which have been collected or which there is access to, and data which is still desired.

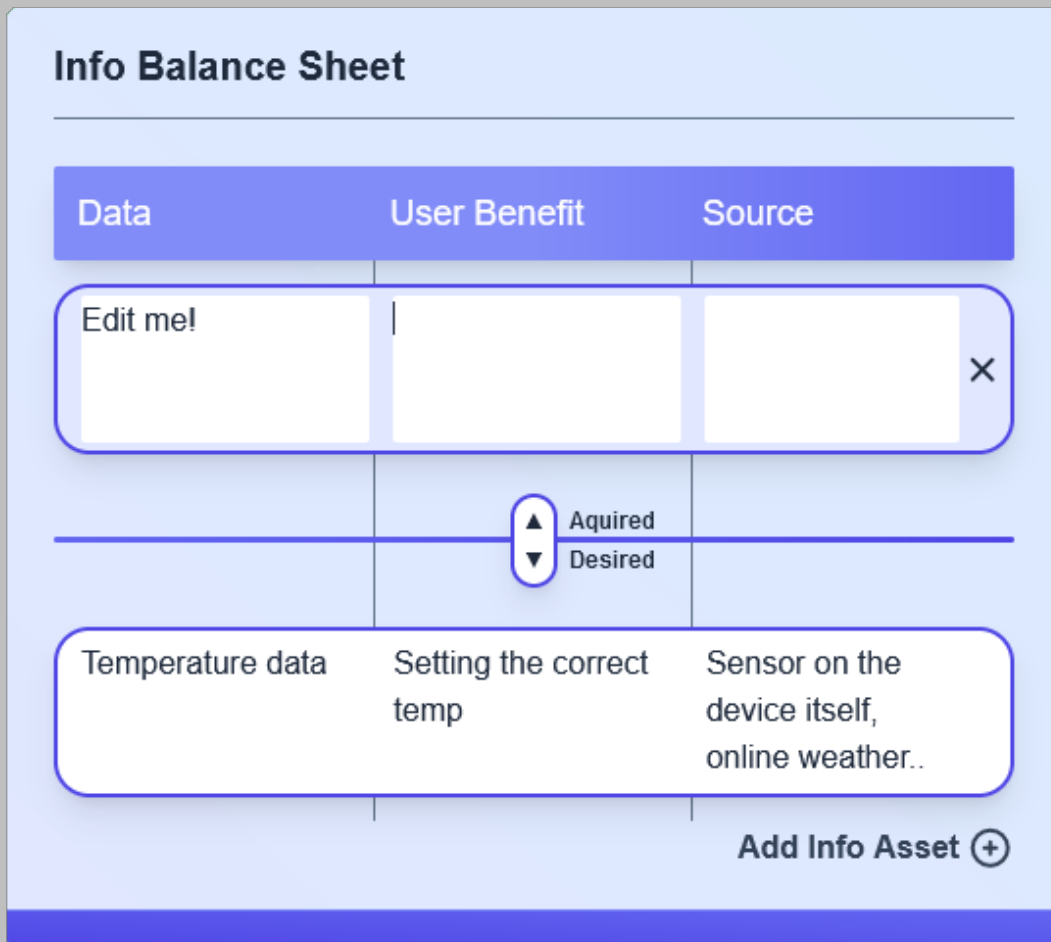
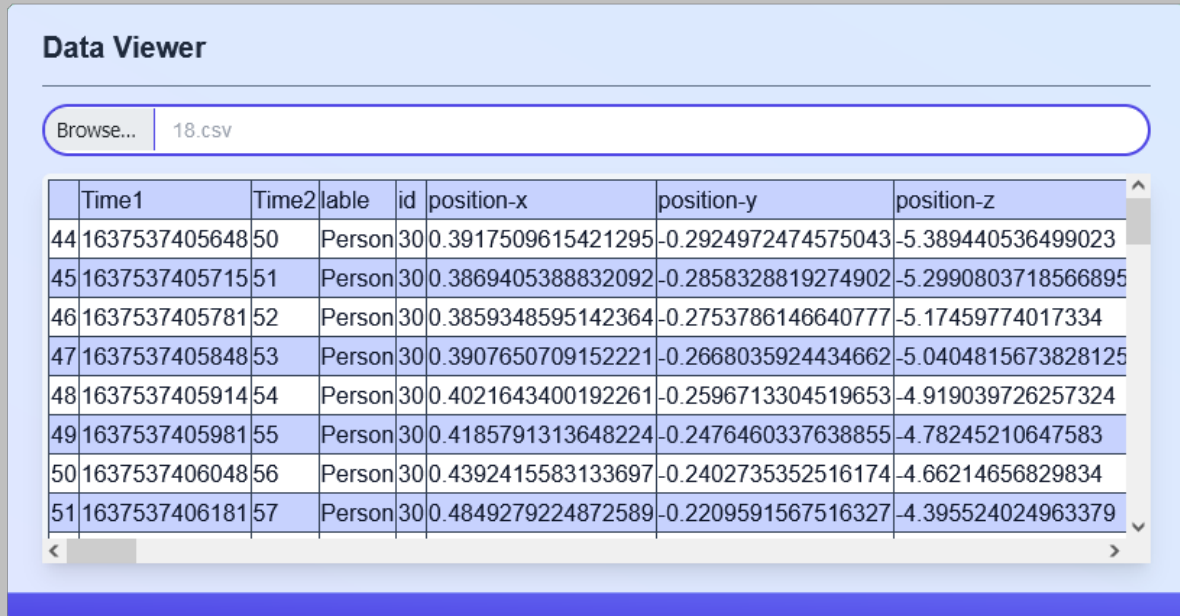


Figure 27: The Info Balance Sheet widget helps users to organize their data types.

Data Viewer

The Data Viewer lets users load CSV data from their device which is then displayed to gain some insights into the dataset. At this time, no other features besides loading are implemented. Outlier detection and basic transformations would be welcome future features.



The screenshot shows a 'Data Viewer' widget with a file input field containing '18.csv'. Below the input is a table with 8 rows of data. The columns are labeled 'Time1', 'Time2', 'label', 'id', 'position-x', 'position-y', and 'position-z'. The data rows are as follows:

	Time1	Time2	label	id	position-x	position-y	position-z
44	1637537405648	50	Person	30	0.3917509615421295	-0.2924972474575043	-5.389440536499023
45	1637537405715	51	Person	30	0.3869405388832092	-0.2858328819274902	-5.2990803718566895
46	1637537405781	52	Person	30	0.3859348595142364	-0.2753786146640777	-5.17459774017334
47	1637537405848	53	Person	30	0.3907650709152221	-0.2668035924434662	-5.0404815673828125
48	1637537405914	54	Person	30	0.4021643400192261	-0.2596713304519653	-4.919039726257324
49	1637537405981	55	Person	30	0.4185791313648224	-0.2476460337638855	-4.78245210647583
50	1637537406048	56	Person	30	0.4392415583133697	-0.2402735352516174	-4.66214656829834
51	1637537406181	57	Person	30	0.4849279224872589	-0.2209591567516327	-4.395524024963379

Figure 28: The basic CSV data viewer widget

These widgets, combined with the reactivity of the benchmarking widget show the direction future developments in the project should take: focusing on reactivity, data and the computational capabilities of the canvas. Using data to actually train models which the app can give feedback on would be the ultimate end goal.

7. **CONCEPT VALIDATION**

Since the greatest personal goal in this project was not just to design but also to create said application, it follows that conducting user tests with the mostly finished application was indeed possible. Initial feedback from experts to a small demo halfway through the prototyping stage was positive but testing with the main target group, design students, was a must.

7.1 Research Questions

When discussing *AILIXR* with students, responses were positive. With the design's feasibility proven in the form of the prototype, the user test conducted at the end of the project was aimed at validating the desirability beyond interest shown in the idea itself. A qualitative investigation aimed to answering the following questions was conducted:

1. Can *AILIXR* (in its current state) be used to supplement ML design courses/design education?
2. Can *AILIXR* (in its current state) be used in ML design projects?

Reviewing the UI/UX was a secondary goal since these are aspects which can be solved quickly as opposed to core design problems.

7.2 User Test Setup

The final build of the project was created for the user test and deployed to GitHub Pages (R. Dekker, 2022b).

The user tests were conducted at the faculty of Industrial Design Engineering with design students with varying levels of ML experience. Since the user tests are qualitative this spread in experience should not matter, in fact collecting insights from more disparate perspectives should be seen as a positive since a greater difference in perspectives should lead to different insights.

The application window was recorded during the user tests as well as audio from the participants using a webcam.

7.3 Testing Procedure

Eight participants were asked to participate in the user test. The study and procedures were explained to the participants after which they filled in the consent form.

Then, the recording was started and the website was opened. Participants were asked to think out loud throughout the user test and to make this easier, the study was done in pairs. Using the app in pairs is likely not much different than the intended use since design is often done

in a team setting. Each participant will have control over the mouse and keyboard for half of the total amount of time.

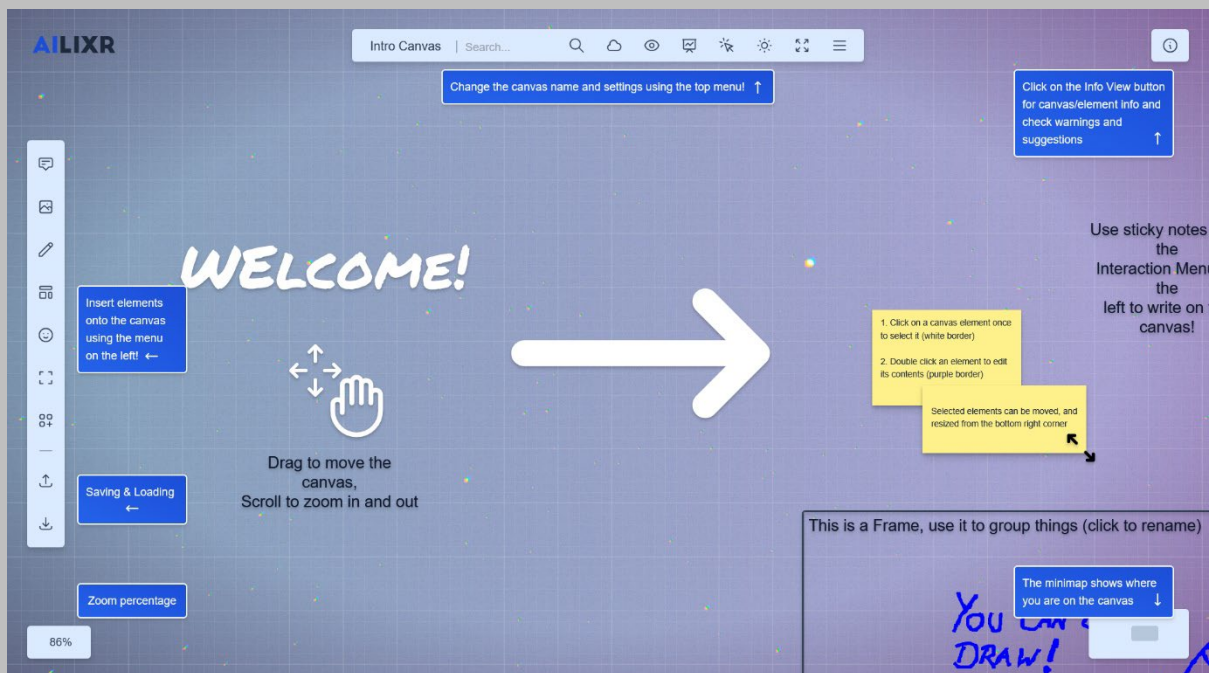


Figure 29: Screenshot from the Intro Canvas.

The first task given to the participants was to open the Intro Canvas, this will give them a quick tutorial on how to use the application (± 3 minutes). Next, they were asked to open and work on the Smart Thermostat case study using the tools provided by the app.

Design Brief

The design brief given in the case study is the following:

Google Nest develops smart home products. You are, as a Delft Designer, tasked with designing the next generation smart thermostat. The original Nest smart thermostat was the company's flagship product. Their next iteration should give users an even smarter experience, benefitting both the user and Google. Smarter energy use is necessary to reduce global warming and should result in reduced energy costs for users, something which is especially welcome now that energy prices have never been higher.

You are asked to design the new thermostat's 'smartness', in other words, its functionalities, features, connections, data, etc. The appearance of the product is outside the scope of the design task.

The new thermostat should be able to:

- Recognize users
- Set and Predict the temperature
- Group user types

Feel free to go beyond these base requirements!

Lastly, the new thermostat will be connected to the internet and can therefore have access to a lot of data, be creative in which kinds of data the device has access to and keep privacy in mind!

The Smart Thermostat case study used in the test is also included in the final design. The three requirements were chosen so that multiple smart systems would likely be embedded in their canvas, each corresponding to one of the three techniques listed in Chapter 3.

When the twenty-minute (soft) time limit is up, participants are asked to rename and export (JSON and PNG) their canvases so they can be analyzed afterwards. The participants were sent their canvases in case they would like to continue exploring the app on their own. Participants are then asked to fill in a form about their experiences with the app in relation to learning about Machine Learning & Design. Two established user experience tools used in the questionnaire are the System Usability Scale (SUS) (Brooke, 1996) and the AttrakDiff scale (Hassenzahl et al., 2003). Recordings and discussions afterwards remain the most important however.

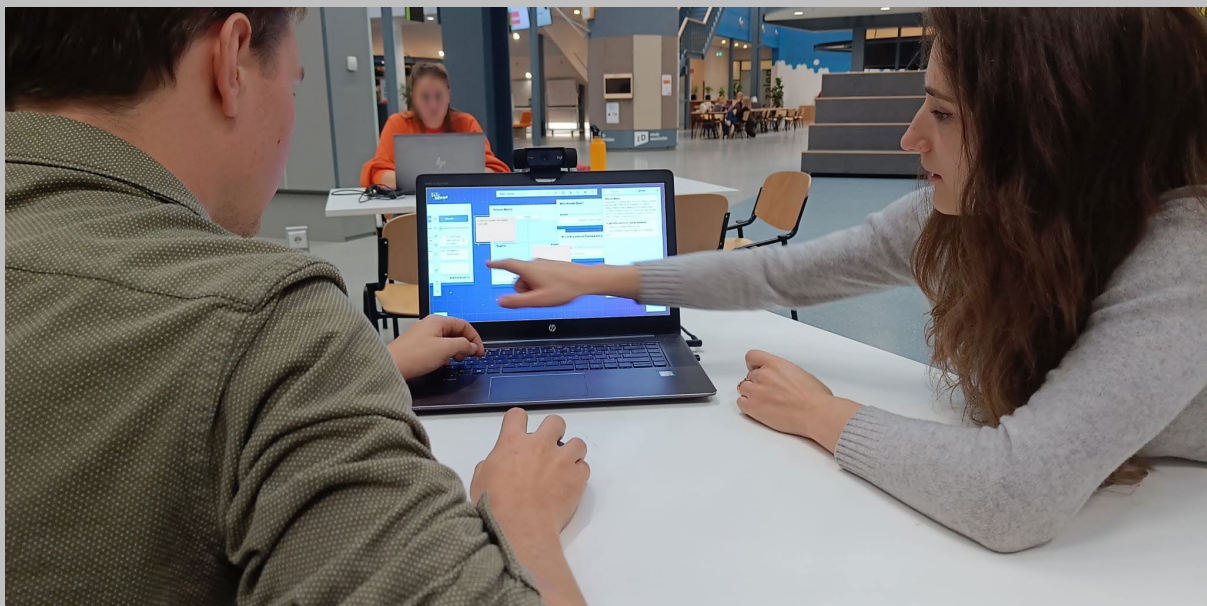


Figure 30: Photo taken during the a user test. Pairing up participants helped in getting them to think out loud.

7.4 Analysis and Results

During the user tests notes of interesting statements and actions of the participants were taken, these can be found in Appendix D. The SUS and AttrakDiff tools yielded quantitative data. Both tools are more related to UX design and do not directly answer the research questions. Still, the user test was an ideal moment to collect UX data which would be relevant for the next phase of the project.

On average an SUS score above 68 is considered above average (Brooke, 2013). The overall SUS score of the prototype was 72.5/100. There are multiple explanations for this: first of all, working with a canvas was not new to the participants as all were design students. Besides this most students did feel that the prototype, though incomplete, was highly polished, multiple stating it looks professional. The results furthest away from the middle of the scale showed that students thought the tool was highly consistent, not too complex and not overly technical. Participants did however not always feel very confident in using the system.

Besides pure usability, the AttrakDiff scale also takes into account many appearance related features. On all 28 aspects, the prototype scored on the positive side of the scale. Features related to appearances scored the highest. Besides these the prototype scored highly on Innovative, Creative, Novel and Manageable. The data gathered using both tools can be found in Appendix E and F.

Observations of the participants using the prototype and the discussions afterwards brought many insights. Besides the plethora of small UX issues (and suggestions for improvements since the participants were all designers), several patterns emerged in all user tests. First of all is that the use of the prototype turned out to be too much app driven instead of design driven. Students did not quite use the app as a tool to support and document their design (process) but simply interacted with what they saw in front of them. The reason for this was the case study which only featured a design brief. The idea was that providing participants with only a design brief would lead them to use this to structure the rest of their interactions. In hindsight, this was perhaps too naive. If all participants would have had a strong foundation in ML perhaps they would have used the ML process to structure their interactions with the app but as it stood, participants explored the app and interacted with anything. From this it can be concluded that the redesigned ML widget menu still does not provide enough structure to guide students towards which widgets are relevant to their process. Part of this failure is due to the user test being the first time the participants interacted with the app and they did not know the full extent of what it had to offer. True though this may be, it does not excuse this

lack of guidance, and the menu therefore needs another round of redesigning. If the case study had a clearer structure with canvas items, UX challenges and widgets already clustered per steps of either the design process or ML process this could altogether be avoided but the tool should be able to be used without requiring an overly structured canvas. Guidance needs to come from the tool itself not the content displayed in it, i.e., the canvases.

Another area where guidance could be improved is the Info View menu. All participants used it when exploring the Intro Canvas, yet when working on the design brief none thought to look at it, even when stuck. Bringing the Info View menu more to the foreground is thus a necessity. The figure below shows what this might look like. Other suggestions were to have the issues and warnings pop up on the widgets themselves or to have the menu slide on screen on its own. It is clear that the useful additional information this menu has to offer needs to be actively integrated, all the while being helpful without being annoying.



Figure 31: The mockup above shows two helpful suggestions: to display the warnings and suggestions below the total number of notifications, and to add chip tags to widget to make it more obvious when these would be relevant to users.

Overall, the results of the user tests were highly positive. All participants stated that they would certainly use the app when designing for ML and they can see it being used in an educational setting on the condition that the information found in the Info View menu is updated.

“It’s a useful tool for designers to learn about the different aspects of AI, because it immediately encouraged me to think about how they affect the users and their experience. By doing the case study and exploring the app’s possibilities, [we] started having meaningful discussions about the use of a given smart product and the importance of privacy, this was encouraged by different widgets and questions in the app.” (user test participant)

“Sometimes during the ML courses we forget about the users, we only have a data focus it seems” (user test participant)

“Normally I don’t know how to synthesize the ML with the design, this tool helps” (user test participant)

In the first test, the users had the expectation that the canvas would save automatically, as a result the participants navigated away and their canvas was lost. They attributed this to the professional look of the app. Another such example is that other participants automatically used the delete shortcut even though they were not told it was a feature.

7.5 Testing Limitations

Multiple factors limited the user test in one way or another. Since the most important part of the user test was qualitative, the relatively small number of participants (n=8) was not an issue. However, the participants did not cover the entire possible user spectrum regarding ML familiarity as no students were intimately familiar with ML. Although the design of the app as presented in the previous chapter was deemed ‘final’, there is still much to be desired, as the testing proved. This was in part the purpose of the user test, to find what was missing, but most participants were able to look past these shortcomings when asked whether the application could be useful in design projects or education. Lastly, the tool was used in a way which more closely resembled a project setting rather than an educational setting which made assessing the usefulness of the tool in this context less optimal.

8. CONCLUSION

This final chapter reflects on the goals set at the beginning of the project and looks ahead to how the project can be continued.

8.1 Meeting Design Goals

Starting with the personal ambitions I had regarding this project I can say I have successfully achieved the reasonable goals I set before starting the project. The outcomes of many graduation projects are conceptual designs which, though substantiated well, fail to be developed further and implemented. With the digital format of the design already decided at the project's conception, it meant the outcome of this project would likely be easier to deploy and implement, especially compared to physical products. My personal interest in web development and programming in general matched this project well and my personal goal to deliver a product which could be meaningfully used at the end of the project was thus set. As argued in Chapter 7, this goal has been reached. However, many improvements are planned as we will see in the following section.

This goal coincides with and fulfills the needs and wants of the primary target group, and not in just a conceptual manner. The Delft University of Technology is the other major stakeholder. Their main concerns are naturally providing quality education to create the engineers of tomorrow. The exceedingly positive responses from the target group means this goal is also met. There are many improvements to be made however. AILIXR is currently not completely suited to be used as a stand-alone educational tool. Of course, this was never the intention, but the following phases will have a greater focus on how to integrate the tool in courses and increase the information available in the app. As it stands, creating workshop and case study canvases can be a valuable supplement to ML design courses.

Goals related specifically to the design were also met. The current version of the application is however only mildly successful. The design first approach which provides a way of working which stays close to design interactions has undoubtedly been implemented in AILIXR. The need for interactivity, as demonstrated by analyzing current tools, has furthermore been met. The extra bits of reactivity give the tools its *raison d'être* over physical or PDF toolkits and provide additional value from an educational perspective. Unfortunately, when critically reviewing the tool's original goal, *closing the gap between ML and design*, the current application is not quite there. In retrospect, the design first approach takes part of the blame. I still fully stand behind this decision but as it stands AILIXR does not bridge the gap as much as I would like. The digital format brings the affordance of computational ability yet the tool does not capitalize on this at all for the time being. The next section discusses how this could be changed.

8.2 Roadmap of the Project's Future / Future Challenges

8.2.1 The Ideal Design

Design can be said to be the art of decision making, of finding the best tradeoffs to satisfy as many wants as possible. It stands to reason then that the final product falls short of the ideal in some way. The constraints imposed by time, my personal knowledge/skill gaps in relation to programming, and realizations late(r) in the project meant that the perfect design has not (yet) been reached. That is not to say the result falls short of its intended goal, on the contrary, results have been exceedingly positive. Seeing as the final design is an application, minor bugs are bound to pop up from time to time. Its digital form is however a strength since applications can easily be updated. The following project roadmap should provide a concrete, clear picture of the continuation of the project.

8.2.2 Technical Roadmap

Throughout the report several points of improvement have already been mentioned. Starting in Chapter 5, a number of technical points of attention were listed. The state of the current prototype does not call for immediate action as it is perfectly functional. Still, starting from the ground up is again necessary for better modularity, DX, performance and general maintainability. To summarize, switching to the Nuxt JS framework and adding Pinia state management for the front end, and adding Supabase as a backend using Express JS as a back end for user authentication and canvas storage. Another reason to start from the ground up is to increase performance so that the app will be able to run well on medium-end devices too. If performance allows, a desktop version could be made using Electron.

Besides the revamped tech stack, the app needs to be deployed and hosted. Currently the app is deployed using GitHub Pages, a free service which hosts static sites from a GitHub repository. Using this method there is less control, the site is static and the domain name ends in .github.io. Switching to a hosting provider would present the site in a more professional way and the improved SEO (another benefit of switching to Nuxt) would enable us to reach a wider audience.

After the app has been deployed, tested and evaluated, phase 3 features can be added. These will be discussed in the following section. From a tech perspective, these require ML models to run on the canvas. There are several options to integrate these: the ml5 and TensorFlow

libraries which are able to run models client side for starters but also external services such as Gradio are an option.

A more in-depth explanation of an updated tech stack can be found in Appendix C.

8.2.3 Feature Roadmap

The previous section already disclosed some of the plans for the future of AILIXR. For now, the intent is to continue working on the project in multiple phases. Counting this project as phase 1, phase 2 is aimed at delivering a product which can be publicly released.

The goal of phase 2 is to rebuild the application and fix the major issues encountered during user testing. The only new features will be user accounts to enable automatic saving and loading of canvases to the cloud. The easier access and sharing of work would add a great amount of flexibility to the tool since users would not be constrained by needing to upload a local canvas JSON file (though this feature does not need to be stripped). A database would furthermore allow case studies, workshop and design brief canvases which are made by the university to be added easily. Easily being able to upload canvases all users have access to would elevate the educational potential of the app enormously. Before publicly deploying the application needs to be tested thoroughly again. Having multiple smaller user tests throughout phase 2 is advised. Before going live, the rest of the website would also need to be designed.

Phase 3 sees the application close the gap between design and ML even further. The app which results from phase 2 is perfectly valuable but there is the possibility to do more. For one, implementing basic data exploration and transformation would offer a concrete way to see data in projects which is essential in ML projects. Lastly, running (pre-trained) models on canvases using different algorithms would be a highly interactive way to explore the possibilities ML offers. Importing data to train models on a canvas would be the next step, however, there are already many tools for creating models. Losing focus by trying to do everything instead of focusing on the design goes against the vision set early on in the project. When the time comes this point may be reconsidered.

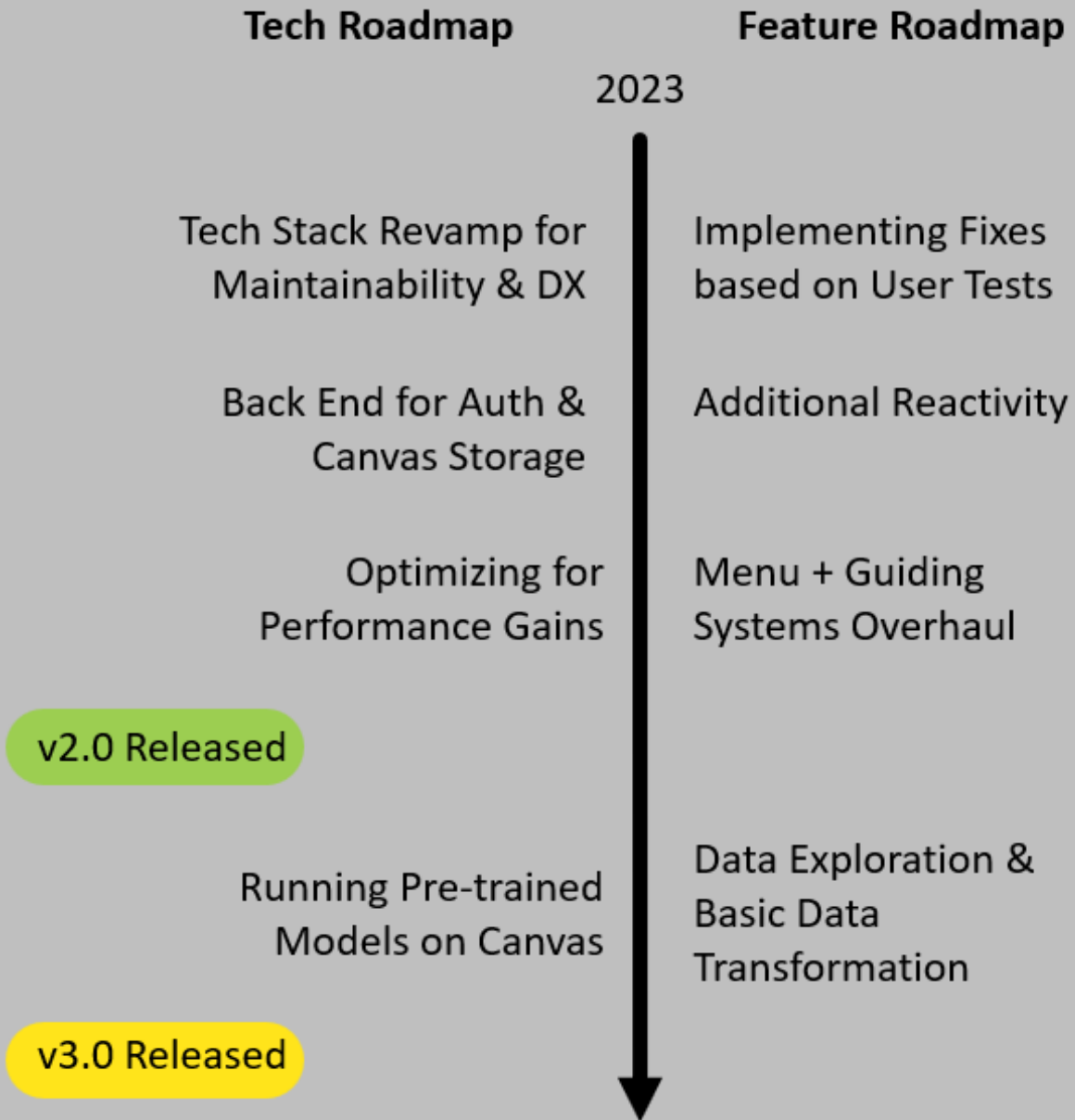


Figure 32: A general overview of the planned features from both the technical and feature side.

REFERENCES

References

A man won an art competition in Colorado using AI-generated art, prompting a debate over what art is. (2022, September 3). Business Insider.

<https://www.businessinsider.com/ai-art-wins-competition-angering-artists-2022-9?international=true&r=US&IR=T>

AL x Design. (n.d.). *AlxDesign Community* | aixdesign.co. AlxDesign.
<https://www.aixdesign.co/>

Allen, M. (n.d.). “*Théâtre D’opéra Spatial*” [AI generated image].

Bauskar, L. (2021, December 2). *Kanban - Visual Studio Marketplace*. Visual Studio Marketplace. Retrieved December 11, 2022, from
<https://marketplace.visualstudio.com/items?itemName=lbauskar.kanban>

Biggs, J., & Tang, C. (2011). *Teaching For Quality Learning At University*. McGraw-Hill Education.

Bloom, B. S., Airasian, P., Cruikshank, K., Mayer, R., Pintrich, P., Raths, J., Wittrock, M., Anderson, L. W., & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*. Longman.

Brooke, J. (2013). SUS: a retrospective. *Journal of Usability Studies Archive*, 8(2), 29–40.

Casakin, H. P. (2006). Metaphors as an Unconventional Reflective Approach in Architectural Design. *The Design Journal*, 9(1), 37–50.
<https://doi.org/10.2752/146069206789372118>

Clark, R. E., & Feldon, D. F. (2014). Ten Common but Questionable Principles of Multimedia Learning. *The Cambridge Handbook of Multimedia Learning*, 151–173.
<https://doi.org/10.1017/cbo9781139547369.009>

Cross, N. G. (n.d.). *The Method in Their Madness; Understanding How Designers Think* [Farewell speech]. Farewell Speech November 8th 1996, Delft, Netherlands.

David A. Kolb. (1983). *Experiential Learning : Experience as the Source of Learning and Development*. *Academy of Management Review*, 8(3).

<https://doi.org/10.5465/amr.1983.4284610>

Dekker, R. (2022a, June 6). *ml-toolkit*. GitHub. <https://github.com/Ruben-Dekker/ml-toolkit>

Dekker, R. (2022b, November 29). *GitHub - GebruikerschNaem/GebruikerschNaem.github.io: MLToolkit*. GitHub. <https://github.com/GebruikerschNaem/GebruikerschNaem.github.io>

Dewey, J. (1986). *Experience and Education*. *The Educational Forum*, 50(3), 241–252. <https://doi.org/10.1080/00131728609335764>

Dorard, L. (n.d.). *Machine Learning Canvas*. OWNML. <https://www.ownml.co/machine-learning-canvas>

Dorst, K., & Cross, N. (2001). *Creativity in the design process: co-evolution of problem–solution*. *Design Studies*, 22(5), 425–437. [https://doi.org/10.1016/s0142-694x\(01\)00009-6](https://doi.org/10.1016/s0142-694x(01)00009-6)

Dreyfus, H. L., & Dreyfus, S. E. (2005). *Peripheral Vision*. *Organization Studies*, 26(5), 779–792. <https://doi.org/10.1177/0170840605053102>

Drops | Free language learning | Available for 45+ Languages. (n.d.). <https://languagedrops.com/>

Elements of AI. (n.d.). *A free online introduction to artificial intelligence for non-experts*. <https://www.elementsofai.com/>

Fastigi, W. (2018, October 14). *What are the biggest myths in education today?* <https://technologyforlearners.com/what-are-the-biggest-myths-in-education-today/>

Freeman, M. (2022, July 6). *Summary Table*. Observable. <https://observablehq.com/@observablehq/summary-table>

Hassenzahl, M., Burmester, M., & Koller, F. (2003). *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität*. *Berichte Des German Chapter of the ACM*, 187–196. https://doi.org/10.1007/978-3-322-80058-9_19

Home. (n.d.). ML4Design. <https://ml4design.com/>

<https://sosmediacorp.com/author/sos-media-corp/#author>. (2018, October 19). *Interactive, Responsive and Reactive – What’s the difference?* Edmonton Web Design, Web Developer & Internet Marketing | Creative Digital Media Agency.

<https://sosmediacorp.com/interactive-responsive-and-reactive-whats-the-difference/>

IBM Developer. (n.d.). <https://developer.ibm.com/blogs/what-is-generative-ai-and-how-much-power-does-it-have/>

IDEO. (2019, July). *AI & Ethics: Collaborative Activities for Designers* | [ideo.com](https://www.ideo.com/post/ai-ethics-collaborative-activities-for-designers). <https://www.ideo.com/post/ai-ethics-collaborative-activities-for-designers>

interact.js - JavaScript drag and drop, resizing and multi-touch gestures for modern browsers. (n.d.). <https://interactjs.io/>

John Brooke. (1996). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation in Industry*, 207–212. <https://doi.org/10.1201/9781498710411-35>

Leufer, D., Steinbrück, A., Liptakova, Z., Mueller, K., & Jang, R. (n.d.). *Home*. AI Myths. <https://www.aimyths.org/>

Manifesto for Agile Software Development. (2001). Retrieved June 9, 2022, from <https://agilemanifesto.org/>

Mcdaniel, R. (2022, June 10). *Bloom’s Taxonomy*. Vanderbilt University. <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>

Metz, R. (2022, September 3). *AI won an art contest, and artists are furious*. CNN. <https://edition.cnn.com/2022/09/03/tech/ai-art-fair-winner-controversy/index.html>

Miller, G. E. (1990). The assessment of clinical skills/competence/performance. *Academic Medicine*, 65(9), S63-7. <https://doi.org/10.1097/00001888-199009000-00045>

ML Toolkit. (2022, November 29). <https://gebruikerschnaem.github.io/>

ml5js-Friendly Machine Learning For The Web. (n.d.). <https://ml5js.org/>

npm. (2022). <https://www.npmjs.com/>

OpenAI. (2022, May 24). *GPT-3 Powers the Next Generation of Apps*. <https://openai.com/blog/gpt-3-apps/>

Piet, N. (2019). *AI Meets Design Toolkit*. AlxDesign. <https://www.aixdesign.co/toolkit>

Pinia. (n.d.). <https://pinia.vuejs.org/>

Puentedura, R. R. (2013, May 29). *Ruben R. Puentedura's Weblog: SAMR: Moving from Enhancement to Transformation*.
<http://www.hippasus.com/rrpweblog/archives/000095.html>

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229.
<https://doi.org/10.1147/rd.33.0210>

Schön, D. (1992). Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems*, 5(1), 3–14. [https://doi.org/10.1016/0950-7051\(92\)90020-g](https://doi.org/10.1016/0950-7051(92)90020-g)

Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (2020, November 15). Tailwind CSS. <https://tailwindcss.com/>

Tailwind Labs. (2022, November 2). *GitHub - tailwindlabs/heroicons: A set of free MIT-licensed high-quality SVG icons for UI development*. GitHub.
<https://github.com/tailwindlabs/heroicons>

TensorFlow. (n.d.). *Create production-grade machine learning models with TensorFlow*. <https://www.tensorflow.org/>

The Intuitive Vue Framework. (n.d.). Nuxt. <https://nuxtjs.org/>

The Open Source Firebase Alternative. (2022). Supabase. <https://supabase.com/>

TU Delft. (n.d.-a). *About IDE*. <https://www.tudelft.nl/en/ide/about-ide>

TU Delft. (n.d.-b). *Delft Design History*. <https://www.tudelft.nl/en/ide/about-ide/delft-design-history-1>

Valamis. (2022, September 28). *Learning Outcomes*.
<https://www.valamis.com/hub/learning-outcomes>

Voûte, E., Stappers, P. J., Giaccardi, E., Mooij, S., & van Boeijen, A. (2020). Innovating a Large Design Education Program at a University of Technology. *She Ji: The*

Journal of Design, Economics, and Innovation, 6(1), 50–66.

<https://doi.org/10.1016/j.sheji.2019.12.001>

Vue Router. (2022). <https://router.vuejs.org/>

Vue.js. (2022). <https://vuejs.org/>

What is Clustering? | Machine Learning |. (n.d.). Google Developers.
<https://developers.google.com/machine-learning/clustering/overview>

APPENDICES

IDE Master Graduation

Project team, Procedural checks and personal Project brief

This document contains the agreements made between student and supervisory team about the student's IDE Master Graduation Project. This document can also include the involvement of an external organisation, however, it does not cover any legal employment relationship that the student and the client (might) agree upon. Next to that, this document facilitates the required procedural checks. In this document:

- The student defines the team, what he/she is going to do/deliver and how that will come about.
- SSC E&SA (Shared Service Center, Education & Student Affairs) reports on the student's registration and study progress.
- IDE's Board of Examiners confirms if the student is allowed to start the Graduation Project.

! USE ADOBE ACROBAT READER TO OPEN, EDIT AND SAVE THIS DOCUMENT

Download again and reopen in case you tried other software, such as Preview (Mac) or a webbrowser.

STUDENT DATA & MASTER PROGRAMME

Save this form according the format "IDE Master Graduation Project Brief_familyname_firstname_studentnumber_dd-mm-yyyy". Complete all blue parts of the form and include the approved Project Brief in your Graduation Report as Appendix 1 !



family name _____
 initials _____ given name _____
 student number _____
 street & no. _____
 zipcode & city _____
 country _____
 phone _____
 email _____

Your master programme (only select the options that apply to you):

IDE master(s): IPD Dfl SPD

2nd non-IDE master: _____

individual programme: - - - - (give date of approval)

honours programme:

specialisation / annotation:

SUPERVISORY TEAM **

Fill in the required data for the supervisory team members. Please check the instructions on the right !

** chair _____ dept. / section: _____
 ** mentor _____ dept. / section: _____
 2nd mentor _____
 organisation: _____
 city: _____ country: _____

comments
(optional)

 |
 |
 |

! Chair should request the IDE Board of Examiners for approval of a non-IDE mentor, including a motivation letter and c.v..

! Second mentor only applies in case the assignment is hosted by an external organisation.

! Ensure a heterogeneous team. In case you wish to include two team members from the same section, please explain why.

APPROVAL PROJECT BRIEF

To be filled in by the chair of the supervisory team.

chair _____ date ____ - ____ - ____ signature Bozzer An

CHECK STUDY PROGRESS

To be filled in by the SSC E&SA (Shared Service Center, Education & Student Affairs), after approval of the project brief by the Chair. The study progress will be checked for a 2nd time just before the green light meeting.

Master electives no. of EC accumulated in total: _____ EC

YES all 1st year master courses passed

Of which, taking the conditional requirements into account, can be part of the exam programme _____ EC

NO missing 1st year master courses are:

List of electives obtained before the third semester without approval of the BoE

name _____ date ____ - ____ - ____ signature _____

FORMAL APPROVAL GRADUATION PROJECT

To be filled in by the Board of Examiners of IDE TU Delft. Please check the supervisory team and study the parts of the brief marked **. Next, please assess, (dis)approve and sign this Project Brief, by using the criteria below.

- Does the project fit within the (MSc)-programme of the student (taking into account, if described, the activities done next to the obligatory MSc specific courses)?
- Is the level of the project challenging enough for a MSc IDE graduating student?
- Is the project expected to be doable within 100 working days/20 weeks ?
- Does the composition of the supervisory team comply with the regulations and fit the assignment ?

Content: APPROVED NOT APPROVED

Procedure: APPROVED NOT APPROVED

comments

name _____ date ____ - ____ - ____ signature _____

_____ project title

Please state the title of your graduation project (above) and the start date and end date (below). Keep the title compact and simple. Do not use abbreviations. The remainder of this document allows you to define and clarify your graduation project.

start date _____ end date _____

INTRODUCTION **

Please describe, the context of your project, and address the main stakeholders (interests) within this context in a concise yet complete manner. Who are involved, what do they value and how do they currently operate within the given context? What are the main opportunities and limitations you are currently aware of (cultural- and social norms, resources (time, money,...), technology, ...).

space available for images / figures on next page

introduction (continued): space for images

image / figure 1: _____

image / figure 2: _____

PLANNING AND APPROACH **

Include a Gantt Chart (replace the example below - more examples can be found in Manual 2) that shows the different phases of your project, deliverables you have in mind, meetings, and how you plan to spend your time. Please note that all activities should fit within the given net time of 30 EC = 20 full time weeks or 100 working days, and your planning should include a kick-off meeting, mid-term meeting, green light meeting and graduation ceremony. Illustrate your Gantt Chart by, for instance, explaining your approach, and please indicate periods of part-time activities and/or periods of not spending time on your graduation project, if any, for instance because of holidays or parallel activities.

start date _____ - _____ - _____ end date

MOTIVATION AND PERSONAL AMBITIONS

Explain why you set up this project, what competences you want to prove and learn. For example: acquired competences from your MSc programme, the elective semester, extra-curricular activities (etc.) and point out the competences you have yet developed. Optionally, describe which personal learning ambitions you explicitly want to address in this project, on top of the learning objectives of the Graduation Project, such as: in depth knowledge a on specific subject, broadening your competences or experimenting with a specific tool and/or methodology, Stick to no more than five ambitions.

Vertical dashed line on the left side of the page.

FINAL COMMENTS

In case your project brief needs final comments, please add any information you think is relevant.

Vertical dashed line on the left side of the page.

Appendix B: Feedback from the Demos given to ML/Design Experts

Participants 1 & 2

Case studies

- Maybe case studies could be thought of as templates (i.e., empty, or slim)
- Each case study may need a little more overview when it opens
- A matrix of widgets vs. case studies
- Need the case study selection cross a range of types - what are the essential categories/archetypes of AI design?
- Think about how case studies are best organized visually and do them all like that
 - o Group widgets with widgets, UX challenges with UX challenges, brief with brief stuff
 - o Or could have more than one version of each case study, organized differently
- Have “blank” versions of each case study
- Think about pre-filled UX widget templates
- One widget could open another relevant one to prompt the user to go to the next/previous step/widget in the design process
- Especially have some bare bones templates - Have a case study open up a blank version
- Create a generic name for the project - like Miro, but maybe MLoD (e.g., Melody where the D is for design — ML Design)
 - o Would help brand it and make it recognizable
- Have other documentation, eventually — like a matrix of different issues to think about as a guide to getting started

- Observable Notebooks: <https://observablehq.com/top>
- Summary Tables for providing a visual overview of a dataset such as from Observable. Example: Summary Table(Freeman,2022).
- . A similar organized view of functions, features, and human + AI abilities/challenges can help you map out new functionalities that you or others can add to your canvas.

Participant 3

- Very enthusiastic about the approach, definitely on the right track
- 2 tips:
- Flow, User flow is not immediately clear, how does the converging and diverging which is part of the creative process work in relation to this concept? Where does that come up?

- User research is a must, definitely ask students. (He helps with the data course in the bachelor's, getting students from there is possible if I contact him!)
- Fermat is an interesting tool but turned out to be totally different
- Vocabulary is a key item. Go beyond the toolkits and put the materials into 'designer speech'. For instance, a confusion matrix could maybe not make sense to anyone who does not have ML experience. Could blend with a known design tool for false positives and negatives.

Participant 4 & 5

- Great first impression but were more critical
- The deck of cards approach is missing but can be added later, it is good that the approach I have goes beyond this.
- A structured template is missing
- They recommended adding frames just like Miro
- Have UX challenges pop up automatically based on which widgets users interact with!!
- Look at how the program SPSS handles getting information on new actions to the user. The program sucks but it does that quite well
- Implications of the context should be clearer
- Extra reactivity and flow are not quite there. It would elevate the concept to the next level, a demonstration of both would hit it home.
- Be sure to highlight more clearly that it is now intended for students but the it is valuable for many more people

Participant 6 & 7

I felt like I couldn't explain the concept as well to them both since they got hung up on the data part when I showed these widgets in the beginning.

- This tool should have more of a data perspective
- Have datasets in the concepts to play around with
- When do widgets play a role and come into play (guiding structure)
- Create more meaningful labels (Vocab just like Jeff said)
- Correspond the program flow with the human centered design cycle
- 'You can't go around the [ML] process', so why is it missing?

Participant 8

- Her first question was again, what/which data?
- See it as having two starting points, either you have data or you don't

- For the slider widgets, make it more obvious that it is a trade-off by having one word grow and the other shrink
- Change model to machine learning model, vocab issue
- Don't think about concepts, think about features
- The first thing you see is highly important when you start the app, what should it be and how should it guide users?
- Flexibility is good, defend it, see it as a possibility. In the end it comes down to managing freedom (I like this paradox)
- Perhaps let users plot themselves on a matrix, familiarity with ML vs position in the process, based on their position the app can give a recommendation on where to start, which canvas template.

Participant 9

- The crux is that for learning ML you need to move to prototyping more
- 9/10 times data is the most important part.
- Give the users multiple pre-trained ML models so they can use them to discover input/output relations
- In practice, smart products use multiple models, my concept is ideal for this
- The connections between components are not explicit but they should be
- Make a clear distinction between what you want the app to have and what you will in fact prototype
- The additive value of Miro is cooperation; it is missing to a degree here. -> make sure the app can run well on a big screen with multiple designers standing around it, touch controls are practically a must
- Don't let yourself be constrained by Miro, look at other E-learning environments (codecademy for instance)
- There needs to be a guiding mechanism to guide the user towards the next steps.

Appendix C - Future Tech Stack

As mentioned before, changing tools during development can be quite tricky. Unfortunately, only towards the end of development, several tools were discovered which would have been a massive aid. One of the problems was the state (data) management within the app since sharing a lot of data between SFCs tends to get more complicated as the app grows in size. Vuex and Pinia are 'store' libraries for Vue, so called because they store state and enable easy access from every component. Changing to Pinia is not straightforward, however, ultimately the structure and clarity it provides would be worth it.

A greater change would come in the form of implementing a different framework: Nuxt 3. This framework is built on top of Vue and features many additional benefits. Features which greatly contribute to a better DX such as: an included router (one which is easier to use than the Vue Router plugin), being able to switch between static site generation and SSR, and the automatic importing of modules, to name a few. Even though Nuxt is built on top of Vue it can't be simply added to existing projects. Luckily, Nuxt and Pinia work well together since both are based on Vue, furthermore, the decluttered development environment greatly improves the maintainability of the project. Although the Nuxt framework was known at the start of the project, the lack of development experience meant that sticking to Vue as a more basic entry point into web development made sense.

Writing software without bugs is an almost impossible task. Testing allows issues to be caught and fixed before release. The larger a project gets the greater the impact of testing (or lack thereof) can be on a project. Other technologies to insert into the tech stack would therefore be testing suites. In general, two levels of testing can be distinguished. Unit testing concerns small parts (units) of code to check its logic and can therefore be as small as a single function. Peeky offers these capabilities with a graphical user interface and is built to work with Vite. Cypress is an End-to-end (E2E) testing suite which is recommended by the Vue core team. E2E testing sits on the other side of the spectrum compared to unit testing and is aimed at holistically emulating the app's real-world use. Since Vue groups functionality within its custom components, their interactions (life-cycle hooks) with the app can be tested too using Cypress or the Vue testing library. Integrating testing suites would also be another learning opportunity.

When looking at the tech stack outlined above, both current and future, most technologies listed so far have pertained to the front end, the exception being Node.js which it is difficult to get around considering its place in the web development scene and its excellent package ecosystem. Express.js (built on top of Node) is a likely middleware candidate. Middleware is

software which manages connections between different parts of a system, for instance, connecting the front end to the back end. Concerning the back end, the development preference regarding which database to use would lean towards Supabase at the moment as it is established and open source. Firebase is also a database option to consider. It is interesting to note that the complete tech stack, including future additions, would fit within the full stack Meteor.js framework, which allows deployment to other platforms besides web and offers other advantages such as app hosting. When the time comes this avenue should be explored further.

Appendix D – Notes Taken During User Tests

Participants 1 & 2

Intro Canvas title was changed because one of the help tags explained this was an option, this of course led to all help tags being removed which is not desirable

“The sticky note looks too much like a comment”

“Dragging items onto the canvas from the menus should be possible”

“Not clear that resizing only works from the right and the bottom”

Benchmarking Widget: Novice would be a better word than layman

Benchmarking Widget: Not all titles are clear

Benchmarking Widget: The information tile should be above the slider

“Background color change is distracting”

In general, the participants started with exploring the tools, not with designing. I got the feedback that it would be better if everything was laid out already.

Value Proposition Widget: Placeholders were not interpreted correctly and could be more clear

“Perhaps display a trashcan when a canvas item is selected, this way you can drag it in to delete it”

The Info View menu was explored in the introduction canvas but was not used at all. “It should pop-up on its own and really draw the user’s attention”

Tension Matrix: The explanation was not clear, there need to be some example values/placeholder values

“The templates need to guide more”

“Perhaps each warning and issues should be displayed on the right side of the screen but underneath each other, the blue dot can still be the total number.”

There was a clear expectation that the canvas would save itself, as a result the participants navigated away and their canvas was lost. They attributed this to the professional look of the app.

“The current app does not teach effectively but is a great tool.” But they could see the app being expanded to remedy this.

Participants 3 & 4

Again, the way the text was formatted makes people want to change the Intro Canvas title which removes the tips

The ML widget menu was too much information at once

“The Intro Canvas needs to work step by step”

They opened the quick setup menu by going back to the home page instead of looking in the menu.

The participants already used the delete shortcut even though they were not told it was implemented. This was “because of the professional look of the app”.

Direct focus is great.

The connections between widgets are missing according to them.

The confusion matrix needs more guidance such as inserting what people want to predict.

“Choosing which widget to use is difficult”

The participants once again took a program/app first approach. They acted following widgets, not the design. The ML process structure is missing.

Participants used each model once thinking that if they used everything they are done.

The formulas should be added to the Precision vs Recall widget.

Benchmarking: It is not clear there are multiple since all marks load on top of each other

The current amount of explanation is not enough

“Perhaps the warning/issues/suggestions should be displayed on the widget itself?”

Nobody understands what JSON stands for, it is not clear this will save the canvas so they can load it in later and continue working. Export should be renamed to Save in the case of the JSON canvas.

Participants 5 & 6

Again, the way the text was formatted makes people want to change the Intro Canvas title which removes the tips

The UX Challenges icon led the participants to believe they would get an emoji or icon menu

Also tried dragging items onto the canvas

“If nothing is selected the canvas should be able to be moved by the arrow keys”

Again, the app was followed instead of the design process.

Again, the participants did not understand what JSON stands for, it is not clear this will save the canvas so they can load it in later and continue working. Export should be renamed to Save in the case of the JSON canvas.

“Perhaps give it a fake extension name which matches the app name, dot MLXD or something”

The Info View was again underutilized.

Info Balance Sheet: The participants wrote everything in one column without being very specific.

Participants 7 & 8

The ML model plot is missing.

Unintentionally activated Draw Mode which led to them being unable to navigate properly. They did not know how to close it.

“How do I know to trust the feedback from the program?”

“The greatest problem is the lack of descriptions in the Info View”

In some instances, there was a language barrier between the app and the participants, more icons, visuals and explanations in the Info View could solve this to a certain extent

“Even though I studied ML I don't trust it”

Again, the process was very much app driven, not design driven.

“Sometimes during the ML courses, we forget about the users, we only have a data focus it seems”

“There should be more tags or keywords” This should help to find the right tools for each situation

“Normally I don’t know how to synthesize the ML with the design, this tool helps”

Appendix E - System Usability Scale (1-5)

1. I think that I would like to use this system frequently	2. I found the system unnecessarily complex	3. I thought the system was easy to use	4. I think that I would need the support of a technical person to be able to use this system	5. I found the various functions in this system were well integrated
4	2	5	4	5
4	2	4	4	3
4	2	4	1	4
3	2	4	4	4
4	2	4	2	4
3	2	4	1	5
4	2	3	2	4
4	2	4	2	4
3.75	2	4	2.5	4.125
Higher is better	Lower is better	Higher is better	Lower is better	Higher is better

Appendix E - System Usability Scale (6-10)

6. I thought there was too much inconsistency in this system	7. I would imagine that most people would learn to use this system very quickly	8. I found the system very cumbersome to use	9. I felt very confident using the system	10. I needed to learn a lot of things before I could get going with this system
1	4	2	4	4
2	4	2	3	2
1	5	1	3	1
2	4	2	4	2
2	4	2	4	4
2	4	2	4	4
1	3	2	4	3
2	4	2	5	2
1.625	4	1.875	3.875	2.75
Lower is better	Higher is better	Lower is better	Higher is better	Lower is better

Sum of (averages of uneven questions - 1) and (5 - averages of even questions): 29	
Total Score: $29 * 2.5 =$	72.5 /100

Appendix F - AttrakDiff Scale

AttrakDiff Scale (Hassenzahl et al., 2003)	Human	Isolating
	Technical	Connective
Participants rated the app on a scale between 1 and 7. Per column, the first word is on the left side of the scale, the second on the right. Averages are displayed below	3	5
	6	6
	6	6
	4	6
	3	5
	3	5
	4	2
	2	6
	3.875	5.125

Novel	Unruly	Pleasant	Inventive	Simple
Ordinary	Manageable	Unpleasant	Conventional	Complicated
2	3	2	3	3
2	5	3	3	5
1	6	2	1	4
5	6	3	3	5
3	6	2	3	3
2	5	2	2	5
2	5	3	2	3
3	6	2	2	2
2.5	5.25	2.375	2.375	3.75

Professional	Ugly	Practical	Cumbersome	Stylish
Unprofessional	Attractive	Impractical	Straightforward	Tacky
4	6	3	4	2
2	6	2	3	2
3	7	2	5	2
2	6	3	5	3
3	7	2	5	2
2	6	3	5	1
1	6	2	3	3
4	4	3	6	4
2.625	6	2.5	4.5	2.375

Predictable	Cheap	Alienating	Brings me closer to people	Unpresentable
Unpredictable	Premium	Integrating	Separates me from people	Presentable
2	4	7	3	6
3	6	3	2	7
5	4	6	5	6
3	5	5	3	6
3	6	5	3	6
3	2	5	4	6
5	4	5	5	5
3	4	6	3	3
3.375	4.375	5.25	3.5	5.625

Rejecting	Unimaginative	Good	Confusing	Repelling
Inviting	Creative	Bad	Clearly Structured	Appealing
6	6	2	4	6
5	6	3	4	6
5	5	2	6	6
5	6	2	5	5
6	6	2	5	6
6	6	2	4	5
5	5	2	6	5
6	5	2	5	5
5.5	5.625	2.125	4.875	5.5

Bold	Innovative	Dull	Undemanding	Motivating
Cautious	Conservative	Captivating	Challenging	Discouraging
2	2	6	5	3
4	3	5	6	4
2	1	6	6	4
3	2	6	4	3
2	2	6	5	3
2	2	5	6	2
3	1	5	4	3
3	2	5	4	2
2.625	1.875	5.5	5	3

