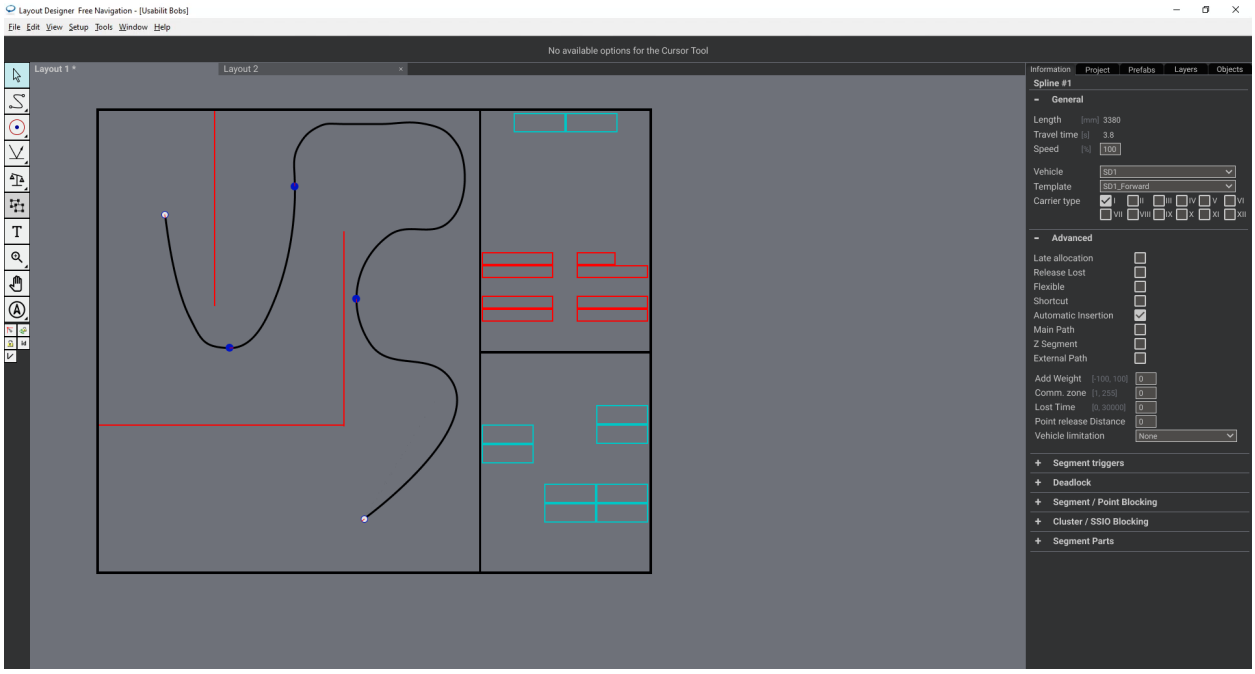


Redesign of Layout Designer

What should be considered when designing for functionality-heavy path drawing software for autonomous vehicles?

Master's thesis in Computer science and engineering

Erik Hildinge & Adam Kjellgren

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Redesign of Layout Designer

What should be considered when designing for functionality-heavy path drawing software for autonomous vehicles?

Erik Hildinge & Adam Kjellgren

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Redesign of Layout Designer
What should be considered when designing for functionality-heavy path drawing software for autonomous vehicles?
Erik Hildinge, Adam Kjellgren

Supervisor: Staffan Björk, Department of Computer Science and Engineering
Advisor: Daniel Gunnarsson, Kollmorgen
Examiner: Olof Torgersson, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Contents

List of Figures

Acknowledgements

We would like to thank our supervisor Staffan Björk for continuous and qualitative guidance throughout the project. We would like to thank Mafalda Samuelsson-Gamboa for sharing her expertise with us when we needed it, despite not being our supervisor. Thank you Olof Torgersson for being a fair examiner and providing valuable feedback on our report. We would like to thank our team at Kollmorgen for their hospitality and their insights about Layout Designer, the subject matter experts who provided us with the knowledge needed for our redesign, and Daniel Gunnarsson who made this project possible and helped us with the communication within Kollmorgen.

Abstract

This project was initiated to redesign an outdated software and explore what to consider during the design of a complex application. It begun with a prestudy, which gathered information about relevant methodology and about the existing software itself. After that process consisted of three iterations of prototyping and then evaluation of those prototype designs for the upcoming prototype iteration. The last evaluation did not result in any further prototyping but rather aided the authors in providing readers with suggestions for continued work. The result of the project was a series of design prototypes and a collection of guidelines to use when performing similar projects.

Keywords: Layout Designer, CAD, Interaction design, User interface, Usability, Evaluation, Prototyping, Redesign, Guidelines.

1

Introduction

Automation is a growing phenomenon that appears more and more around the world. It is present in people's everyday life, e.g. as robot vacuums and there are already some self driving vehicles out in the open world as well [1]. Some of these robot vacuums has an initial setup phase, in which they map their environment for more efficient cleaning compared to when they do not have their layout mapped [2]. In a similar fashion the automatically guided vehicles that *Kollmorgen* develop software for has an initial setup phase. A software called *Layout Designer* is used to manually draw paths, on top of a Computer-aided design (CAD) drawing of the area, for the vehicles to follow. These layouts have to be very accurate as errors or deficiencies in the layouts can lead to poor performance. No major updates of Layout Designer has been done since the main development in the late 90's. Due to this, the software lacks relevant design patterns such as Safe Exploration [3, p.12-13], many actions are hard to undo and therefore an error might lead to the user having to start over from the latest saved version of the layout. As a result of the aforementioned difficulties, designing a layout for the automatically guided vehicles can become a frustrating and time consuming task.

As the usability is the main problem with Layout Designer, the research area will be just that, usability. The scope will however be somewhat more narrow, to evaluate usability within path drawing software's (such as Layout Designer). As a result of this the research question to be answered is "**What should be considered when designing for functionality-heavy path drawing software for autonomous vehicles?**". The answers that we are expecting to get is in the form of guidelines, which should be considered when designing programs that are similar to Layout Designer.

The design process will be done in iterations, with a lot of emphasis on feedback from expert users. Since this is a complicated software with a lot of features, it is important to be aware of which features is the most prevalent and design accordingly.

1.1 Limitations

It is not within our scope to integrate the suggested changes, this is in order to efficiently explore relevant alternatives and not waste precious time.

A factor that limited parts of this project was the outbreak of the Corona Virus (COVID-19) [4] which resulted in a significant part of our project being executed remotely.

1.2 Reading instructions

While the report gives the most insight while read as a whole, researchers and students will probably be the most interested in Chapter 6, 7, 8 and 9. Interested parties at Kollmorgen will most likely be the most interested in Chapter 7, 8, and 9. Every chapter in the Appendix is also of value to Kollmorgen but Appendix B has the most value for further improvements of Layout Designer.

2

Background

One of the products that Kollmorgen provides is a steering system for automatically guided vehicles, with correlated control solutions [5]. The vehicles are guided by a set of instructions and a path to follow drawn in Layout Designer. This software has not been subject to any major updates in the two most recent decades, which results in a lack of new usability findings being implemented.

Layout Designer has an acceptable performance for the experienced users at Kollmorgen, but even they can make errors which are hard to correct due to how Layout Designer currently works. However, the main problem is its lack of usability focused on new users. Visibility of the system status [6] is one problem Layout Designer often displays, as a new user it can be hard to predict where the drawing might end up when you scroll or zoom, as Layout Designer does not update the view until the user has stopped any action. Undo- and redo-functionality are present *sometimes*, but not always, which makes it lack both user control and freedom [6], and consistency [6].

The person who made Layout Designer is no longer working at Kollmorgen, but it can be assumed that the program had a much different software development process than is suggested today. Back in the earlier days of software development, it was not unusual for managers to initiate the process by capitalizing on market opportunities by realizing product requirements, which then developers both built and tested [7, p.8]. This process is still evolving, but a suggested alternative (which should be iterated upon) is for managers to mandate the idea to designers, that give specifications to developers, who then hands the application over for testing, and this process goes back and forth until the product is ready for shipping [7, p.8]. Obviously, a lot of steps were missed during the development of Layout Designer, which may explain a lot about the flaws of the application.

2.1 Competitors

Kollmorgen has competitors regarding their drawing software, but also within the field of Automated Mobile Robots (AMR), and the robotics industry (which includes the steering software for the trucks). The companies that have a similar drawing

software to Layout Designer does not allow distribution of pictures.

Two companies with very relatable softwares to Layout Designer are *Navitec* and *Bluebotics*. Both of these softwares have their basis in drawing the specified paths within a CAD environment. Bluebotic's software allow for the user to draw walls within the program, while Layout Designer needs to have a layout imported (very specifically within the positive X-, and Y-axis). Navitec has a similar appearance to Layout Designer, with some more updated graphics for the trucks. Since these softwares are not free to use, it is difficult to determine absolute advantages and disadvantages.

Competitors within the robotics industry are companies such as: *Aethon* which focus on healthcare, hospitality and manufacturing [8], *Cobalt Robotics* that focuses on security and facility management [9], and *Fellow AI* that provide robots that manages in-store stock [10]. Kollmorgen do not make the robots themselves, but rather just the software that controls them.

2.2 Stakeholders

This thesis will be written as part of the Interaction Design and Technologies Master's Programme at *Chalmers University of Technology* by *Erik Hildinge & Adam Kjellgren*. It will be made in collaboration with the Mölndal based company Kollmorgen, which requested this research to improve one of their programs. The users of the program as well as Kollmorgen's potential business partners are also considered to be stakeholders.

Kollmorgen is a company that previously was involved in designing and manufacturing periscopes for submarines in the early 1900's [11]. Kollmorgen is one of few companies that manufacture and design classic servo motors; direct drive servo motors; stepper motors; drives & amplifiers; gearheads; actuators; or CNC & multi-axis motion controllers, while also being a leader in AGV solutions in both hardware, and software for navigation technologies [12].

2.3 Inspirational programs

Some programs that was easily relatable for most people being shown Layout Designer was Adobe Photoshop and Microsoft Word, not because they were necessarily extremely similar in function but rather has different sorts of functionality that they manage to handle very efficiently. In the approach towards redesigning Layout Designer, it will be important to keep these two programs in mind.

2.3.1 Adobe Photoshop

Adobe Photoshop is, according to themselves, the best photo, image, and design editing software [13]. Most people probably recognize it for the ability to perform

quite advanced editing on pictures, even when being a novice in the program.

The program has a lot of different functionality that is intuitively learned and adapted by the user, with a combination of key-bindings, clever button placement, and relevant icon representation. It is entirely usable by new users for the simpler tasks that they want to execute, even though Adobe Photoshop dedicate themselves towards the intermediate- to expert users [3, p.4].

2.3.2 Microsoft Word

Microsoft word is a word processing program that is commonly used to create documents, this program is used by novice as well as expert users.

A majority of the available functionality is available in the top toolbar, shown in Figure 2.1. With the exception of modeless feedback regarding: the page number, the number of words, and the current language. The bottom bar contains tools that allow the use to change the *page mode* and *zoom* in or out.

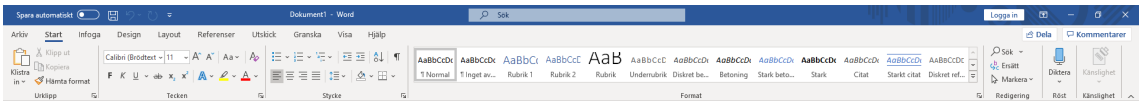


Figure 2.1: The top toolbar of the current version of Microsoft Word

2.4 Layout Designer

As the appearance of Layout Designer most likely is unknown to most a picture of the Software can be seen in Figure 2.2. In order to not disclose too much information that would breach contract, all of the functionality will not be shown unless necessary.

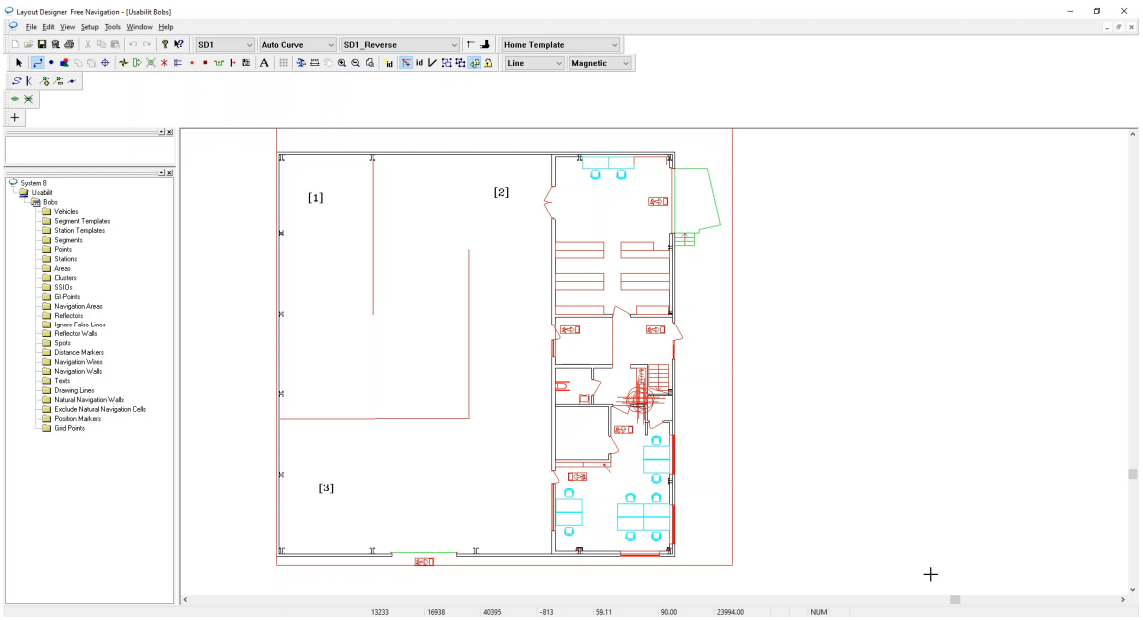


Figure 2.2: This is how Layout Designer looks while having a layout open. The visible layout does not represent a realistic factory.

3

Theory

In this chapter research related to the domains of interaction design and software similar to Layout Designer will be presented. General design principles and information about how to design for people are also presented.

3.1 Related Research

In this section research related both to the domain of Layout Designer (a CAD based software used for automation) and the domain of interaction design will be presented.

3.1.1 Research Related To Layout Designer

Amit Kumar Bedaka and Chyi-Yeu Lin have written a report, regarding a CAD-based robot path planning and simulation software which uses OPEN CASCADE, in which they claim that human operators are not cost effective nor time efficient in manufacturing processes [14]. Some applications in manufacturing requires human operators, but some things can be carried out by robots. A common method to program the robots is by using a *teach pendant*, but modifications of the robot path are time-consuming with this method according to Bedaka and Lin [14]. They also claim that their CAD-based human robot interaction platform is cheap and user friendly. In a short time span, a few minutes, a user was able to create a path to execute a task, as well as load the instructions to the robot [14].

Ghange Lee *et al.* has evaluated complex engineering applications, with focus on CAD in 3D, and presented some best practices [15]. A best practice for drawing generation is that the drawing and 3D model has a bidirectional relation, i.e. changes in either one are automatically applied to the other [15].

In a majority of the evaluated CAD systems help was offered to the user, either by help files or automatically provided depending on the current action [15]. Help files were problematic due to steps or information missing, many users reported, help files which contained animated tutorials were very effective in compensating files that had missing steps [15]. Lee *et al.* judged the hierarchically structured help

files as good, another approach, providing an information category and a tutorial category for each concept was judged to be effective as well [15]. They have divided help related to the current context, *context-sensitive help*, into thee categories: Pop-up tooltips, prompts for next step at a static point in the program, and prompts located close to the mouse. Context-sensitive help from all categories was able to aid users when executing operation sequences. They claim that prompts located close to the mouse require less eye travel from the user [15].

Lee *et al.* believe that categorising top level menus, either by relevant work phase or the type of object can help users locate functionality. They also state that it is generally helpful to allow the user to customize their menus in complex applications [15].

Lee *et al.* state that screen real estate can become scarce since complex 3D parametric modeling systems often have so many tools that they can not be displayed all at the same time. According to them CAD systems handle this in three different ways: grouping, collapsing toolbars, and modes. The example they give for grouping is that one icon can be clicked to display a group of similar actions, all represented by the first icon. The collapsing toolbars allow the users to choose when to show toolbars and when to hide them, Lee *et al.* do however argue that this solution might be inferior since users can have a hard time finding tools that are part of the collapsed toolbars. Modes are a possible solution when icons have logical categories, they also found systems where the modes were based on user experience. This can allow expert users to use modes with fewer visible tools as they often prefer to use keyboard commands [15].

When opening models that have been created previously, or creating new models Lee *et al.* found that the need of learning can be reduced by adhering to the operating system's conventions [15].

Regarding selection direct manipulation was generally the method found by Lee *et al.*, either by clicking edges, clicking surfaces, or drawing a bounding box around the desired object(s). Highlighting and color coding were often used to distinguish the selected objects from the ones that had not been selected, according to Lee *et al.* [15].

Lee *et al.* examined a few applications which had automatic checking built into the software, to make sure specific actions are valid while being performed. Some applications had warnings for situations that could potentially lead to problems or if it was bad design [15].

Context-sensitive access to commands was present in some applications, this allows the user to use relevant functionality based on what is selected at the moment. The access to this functionality could be mouse right-click or keyboard shortcuts. This type of functionality access allows expert users to work faster than they would otherwise, according to Lee *et al.* [15].

The views in the programs Lee *et al.* could be manipulated, displaying transparency, colors and other visibility related traits. In all CAD applications it is important to be able to control visibility, this can be done by toggling visibility of specific layers. Usually this functionality is accessed via a dialog box [15].

From the observations Lee *et al.* made of the different applications they compiled best practices for different problems within user interface areas. They divide these practices into overarching categories, *Principles for general system design. Principles specific to 3D parametric design*, and *Principles for user support* [15].

Regarding general system design Lee *et al.* advocate for consistency, making methods of interaction consistent between operations that are similar to one another. They suggest that relevant information should be clearly visible to the user. They suggest that the application should provide immediate feedback to inform the user of the application's status. The application should provide the user with necessary functionality to recover from errors, this could be undo and redo commands [15].

Regarding the 3D parametric design Lee *et al.* suggests the primary functions of the CAD application should be given a big portion of the screen space to maximize the workspace. They suggest that when it is appropriate user comprehension can be enhanced by using graphical information instead of textual information. Direct Manipulation should be provided, i.e. the user should perceive interaction as having an instantaneous result in the application [15].

To support the user Lee *et al.* advocates for familiarity within the application, make use of the experience from other applications or interactions in the real world, such as having similarities to the conventions in the operating system. Users should be able to modify the interface to align with their own preferences. The users should be assisted, both in a detailed manner and implicitly while working. This assistance can be detailed in help files and tutorials and implicit assistance can be provided via tool tips. They suggest that the design should be kept simple, and to not provide inessential information as it may be the source of confusion. The example given to achieve this is to hide or disable options in dialogs or menus when those options are not valid. They also suggest that the context should be recognized by the application, to automatically adjust the interface based on the current mode, as well as providing context-sensitive help [15].

Ingrid Pettersson and Wendy Ju describes several exploratory interaction design techniques that could be useful within the field of human vehicle interactions, where the techniques used and described are more specifically *Wizard of Oz, enactment, scale scenarios, contextual inquiry, field experiments, and video and animation prototyping* [16, p.147]. The paper intends to make the user understand how to design automotive interactions using an interaction centered approach with the focus of developing a trust between the user and the technology [16, p.156].

Pedro Neto *et al.* describes the advantages and disadvantages of applications using

robot simulations within a CAD environment, such advantages may be to simulate the robotics in a manufacturing process, without it necessarily existing in a physical form. It is also useful in the process of evaluation, in order to estimate the performance of the robots in a physical environment, and could also make the robot programming easier. The examples provided are set in a 3D environment, and are not suggested to be tried in a modern manufacturing system, but should rather be implemented in smaller enterprises and for educational purposes. Emphasis lies within the off-line simulation systems, which makes it possible to test the robot programs without interfering with (and slowing down) the production. [17, p.333].

3.1.2 Research Related To Interaction Design

According to Stanley Dicks the concept of usability testing is under risk of becoming undermined if the usability testing methods are performed by people who do not understand the methods. He states that mis-usability is contributed to by a few groups of misconceptions. Inadequate understanding of usability, statistical problems such as misusing statistical results, not using usability tests for usability, limited understanding of limitations and methods used in usability testing, and not testing for usefulness [18]. Dicks mentions that although not all problems are discovered when tests are performed with small sample sizes, this can, to an extent, be ameliorated with an iterative testing process to find a majority of them.

Joseph S. Dumas and Janice C. Redish claims that five characteristics are present in every usability test even though a usability test can be conducted in a wide variety of different ways. Those five characteristics are the following.

- Improving the usability of the product is the main goal.
- The participants in the test represent actual users.
- Real tasks are performed by the participants.
- What the participants do and say is observed and recorded.
- Real problems are diagnosed by analysing the data, recommendations to rectify the aforementioned problems are made.

Dumas and Redish mention that the tasks the users are asked to do must be tasks they will actually perform when using the product, in other words it is important to understand which tasks that are relevant. It is often the case that only a portion of the many existing tasks can be tested, especially in software that offers a lot of functionality [19, p.22-23]. Dumas and Redish adds to this, the usability test is only successful when it does help improve the tested product [19, p.25].

Dias and Paiva writes that a poor user experience can make users pick another program, therefore usability is important in software [20]. They claim that some aspects of usability can be tested completely without users and with automated tests, called Pattern-Based Usability Testing (based on Pattern-Based GUI Testing) [20].

3.2 Design Principles

Generally, *About Face* by Cooper *et al.* [7] is a good source of design principles, and many of the coming quotes and guidelines are taken from there with a reference to the corresponding page.

3.2.1 Unnecessary confirmation

Something that is overwhelmingly present in Layout Designer is the different occasions where a modal warning popup with the only option of pressing OK shows up. Cooper *et al.* describes that most of these popups are only irritating and demeaning for the user since it is not always obvious what went wrong and it is still not OK that the error occurred [7, p.5].

Tidwell suggests that modal panels should only be used while the application can not or should not proceed without getting input from the user. An example given is when trying to save a document that is not yet named [21, p.97].

3.2.2 Disrupting workflow

Cooper *et al.* discusses how even the most advanced applications can have discrepancies in their workflow [7]. In a quote regarding the Adobe Photoshop palette controls, and how the user is forced to navigate between the Bucket and Gradient tools since they both occupy the same location on the tool palette, he says the following. "... However, both are fill tools, and if both are used frequently, it would be better to place each of them on the palette next to each other to avoid that frequent, flow-disrupting tool navigation."[7, p.276]. So even though Photoshop may be a great source of inspiration in many cases, it is important to consider that the workflow may not be as optimal as possible.

3.2.3 Failing to meet user goals

Cooper *et al.* mentions four key factors that describes how user interfaces fail to meet user goals [7, p.13]. It is because they:

- Make users feel stupid
- Cause users to make big mistakes
- Require too much effort to operate effectively
- Don't provide an engaging or enjoyable experience

Tidwell mentions during the explanation of *Safe Exploration* that if people have the possibility to explore the application without any setbacks or repercussions they are likely to learn more and have a positive experience. Tidwell simply explains it with "Good software allows people to try something unfamiliar, back out, and try something else, all without stress."[21, p.9]. An interface that allows safe exploration would ultimately be successful in meeting a lot of the user goals.

3.2.4 Implementation-, Mental & Represented model

The **implementation model** directly translates to how the technology or machine actually works, where one button for one function is a good example. It is generally much easier to design after the implementation model since it is more logical from the developer's perspective [7, p.17].

A **mental model** is also commonly referred to as a *conceptual model*. This model represents how the user imagines that the machine or application works. The user does not need to know and does not care about how the application *actually* works, they only want to use it [7, p.17].

The **represented model** is primarily something that is present in software since an interface is generally implemented in order to represent the functions of the software. Since there is an off-set between what is visualized and what is implemented, this gives birth to the represented model [7, p.18].

It is described by Cooper *et al.*, that an application's implementation is supposed to be as close to the mental model as possible: "The closer the **represented model** comes to the user's **mental model**, the easier he will find the application to use and understand."[7, p.18].

3.3 Design for people

Design for people consists of four parts, *context, goals, research, and the patterns*. To find the context it is important to know who the audience is, the goals are what the audience is trying to accomplish. The research refers to ways to understand the aforementioned context and goals. The patterns refer to usual thoughts and behaviours people express in relation to software interfaces [3, p.2-11].

Tidwell *et al.* states that the content and functionality should be matched to the audience, and if the users are not proficient with computers complicated widgets and unusual design conventions should be avoided [3, p.3]. Software programs can be designed with the users' skill level in consideration. Some software, such as *Photoshop* [13] and *Excel* [22], is designed for intermediate or expert users, while the opposite is true for installation wizards and the purchase pages used by online stores, they instead want to accommodate the occasional users. A majority of all applications is in the middle of these two, the applications want to accommodate new users as well offer a smooth experience to the frequent users [3, p.4-5].

4

Methodology

In this chapter several methods are described. Not all of the described methods will be part of the final process.

4.1 Research Through Design

In *research through design* design researchers should focus on making artifacts move from the current state to the state that is preferred [23]. Zimmerman *et al.* expects design researchers to de-emphasize economical aspects to be able to prioritizing making the correct thing, rather than a commercially successful thing. These contributions should, according to Zimmerman *et al.* not be a refinement of an already existing product but rather an artifact that can display notable invention.

William Gaver suggests moderation of expectations on research through design's ability to produce verifiable theory [24]. According to Gaver design is often used to adress wicked problems[24], a problem that can not be described in detail before having knowledge of conceivable solutions [25, p.136-137]. Gaver states that design is generative, instead of confirming *what currently is* design is focused on created *what might be*, which he connects to the formulation *making the right thing* which Zimmerman *et al.* makes use of. It is therefore not in line with research through design's methodological approach to attempt to disprove, the goal should be to produce theories that are correct in some instances, the goal is not to produce irrefutable theories [24].

The main goal of Designing through Research is to build knowledge in order to enhance the design practice. To start the process, it is necessary to look at secondary design research and combining that with exploratory research. Recommended methods for the exploratory research include the previously mentioned: Contextual Inquiry (see section 4.8), Observation (see section 4.16), and Interviews (see section 4.13) [26, p.146]. The result after ideation, experimentation, and critique should be a more correct solution than the previous one. Documentation is important when iterating though solutions, so a reasoning behind the design choices are clearly communicated [26, p.146]. Research through Design can be used during *all* design stages [26, p.5-7].

4.2 Social Research

In *Social Research* the process to find answers to how something is, and why it is that way, is applied to people [27, p.9]. To find an answer to these questions it is important to state what is already known about the situation and what the goal is, to be able to state what is not known and what the research can be used for [27, p.13-14]. According to Yoland Wadsworth social research should not be unreasonably more time consuming than the processes used to find out things in everyday life. Wadsworth suggests keeping the research project small [27, p.42]. Wadsworth identifies four common mistakes regarding timelines [27, p.36].

- A planning phase that is too short.
- Spending too much time on collecting data
- Assuming that summarising the data will be substantially quicker than it actually is.
- Not leaving enough time to actually produce something from the gathered information.

4.3 Goal-Directed Method

Cooper *et al.* describes the difference between goals, activities and tasks, as goals being driven by human motivations which change very slowly (or not at all) over time, while activities and tasks are almost entirely reliant on whatever technology is at hand. The Goal-Directed Method designs with the goal in mind, rather than the activities and tasks.[7, p.15]

According to Cooper *et al.* goal-Directed design can answer several important questions when designing a digital product, such as "How will my product help infrequent and inexperienced users understand how to accomplish their goals?", "How will my product help infrequent and inexperienced users understand how to accomplish their goals?", and "How can my product's functions be most effectively organized?".[7, p.29]

Cooper *et al.* states that "While in the process of ideation, it's important for designers to have a clear mandate for moving forward. While the Goal-Directed method aims to define products and services via personas, scenarios, and design requirements, it is often useful at this point to define in which direction these scenarios and requirements should be headed."[7, p.110].

Cooper *et al.* mentions a couple of suggested activities to effectively conduct qualitative research. The suggested activities (in this order) goes: Kickoff meeting, Literature review, Product/prototype and competitive audits, Stakeholder interviews, Subject matter expert (SME) interviews, User and customer interviews, and User observation/ethnographic field studies. The order of these activities may depend on the situation of each project.[7, p.36-44]

4.4 Affinity Diagramming

Affinity Diagramming can be used both for contextual inquiry and for usability tests. When used for contextual inquiry each observation from interviews should be written on its own sticky note. When the notes have been placed, preferably on a movable surface if re-positioning is necessary, they can be interpreted and rated. Notes with similarities, such as the intent or underlying problem, are grouped together. Information about the users, what they try to accomplish, and what their problems are can be interpreted from these groups [26, p.12]. When Affinity Diagramming is used for usability tests, each participant in the test is represented by one color of the sticky notes. Observations are written down during the tests, when grouping the notes common problems will be represented by many different colors, indicating that multiple test participants experience a certain problem [26, p.12]. Both variations have their groups made by first placing related notes together to create themes, not by having predefined groups to place notes into [26].

4.5 Cognitive Walkthrough

Cognitive walkthroughs can be used to evaluate if it is easy to learn an interface. It can be especially good for testing interfaces where the user is most likely to be a one-time user, or a first-time user [26, p.33]. The method is appropriate when evaluating "walk-up-and-use" systems. To evaluate a system, each step the user takes while trying to accomplish a certain task can be evaluated to assess if that particular step moves the user away from or towards the intended goal [26, p.32].

4.6 Competitive Testing

The process of evaluating the usability of competitors' products is Competitive Testing. Monitoring company size, revenue, and operating profit is also part of this process even though it is rarely considered for user-centered development [26, p.36]. When Jakob Nielsen analyzed studies in which Competitive Testing was used he found that the average difference in usability to be 68% when competing companies were compared [28]. When performing competitive testing, attitudes towards competing products are not to be considered, focus should instead be on the behaviour of the end-user when they perform tasks available in all products. The company's own product as well as three to four competing products should be part of the usability tests. It is equally important to find similarities and differences between the company's own product and the competitors' product, this can in turn be used as clues for specialization of the own product [26, p.36]. Best practice when doing competitive testing is to not reveal the own company's name as to avoid introducing bias, it is worth considering hiring a third party to perform the tests to avoid complications [26, p.36].

4.7 Contextual Design

Contextual Design is built up from multiple steps, the recommended ones are the following, in order.

- Contextual Inquiry
- Interpretation Sessions
- Work Models and Affinity Diagrams
- Visioning and Storyboarding
- User Environment Design
- Paper Mock-ups

To perform the contextual inquiry (see section 4.8 for a more thorough description) you talk to customers, to find out what they need in order to perform their job everyday. The interpretation sessions are to analyze the user data from the interviews in a structured matter. Both work models and affinity diagrams serve the same purpose, to help teams show the intricate systems through an externalized representation. To create new concepts regarding how the customer's work can be improved, visioning and storyboarding are used. The user environment design is in turn used to represent a new design of the system that can improve the current behaviors and support the customers' natural course of action during their work. User environment design should also document the system's structure, its functionality, and the flow. The paper mock-ups are used before implementing the proposed design, to get customers' feedback on the new structure, function, and flow. [26, p.43-44]

4.8 Contextual Inquiry

Contextual inquiry helps structure interviews to reveal relevant information about how the participants work, as when unobserved, people might miss these details in their summaries of the tasks and processes they go through [26, p.47]. The method of contextual inquiry is defined by four principles, context, partnership, interpretation, and focus.

The researchers should be on site, details about the relevant people and their activities has to be observed to develop an understanding of the whole experience. This is how the context is integrated into the contextual inquiry and is the most basic of the requirements [26, p.46].

The knowledge transfer of an individuals work structure is more reliable when people can explain how the work is done simultaneously as doing the work. This is what is called the partnership, as an apprentice can learn by observing the master, the master can teach the apprentice by demonstrating. This will in turn lead to a better representation of reality within the gathered data [26, p.46].

The collected data, answers to questions, statements, and observations, has to be

interpreted to understand what design implications it can lead to. The interpretations must be double checked with the participant while still at the work location, as to not have any misinterpretations that can later lead to unsuccessful design implications [26, p.47].

The researcher's focus must be on the participants and their world. If for some reason the researcher is surprised, or thinks a participant is acting in a peculiar way, or behaving in a contradicting manner the researcher is given possibility to refocus to learn more about the participants [26, p.47].

Contextual inquiry can be used to help the researcher understand the flow of communication, the task sequences, how work is accomplished through usage of different tools, how culture can influence the work, and how the physical environment can have an impact on the work as well [26, p.47].

Two to three hours is a common duration to execute a contextual interview in. The number of interviews needed is affected by the project's scope and what the result is intended to support. In all cases it is important to interview multiple people before starting the grouping of contextual inquiry findings [26, p.47].

4.9 Critical Incident Technique

Designs for future users can be optimized if one understands how the company's products are experienced by users during critical moments, for this the critical incident technique can be used [26, p.50].

When an event occurs and the anticipated result differs from what actually happens a *critical incident* has occurred. This type of event can be designed for with the help of the critical incident technique, abbreviated CIT [26, p.50].

With CIT the researchers ask participants about descriptions regarding situations ending in a way they perceive as either positive or negative. These experiences can be collected either via directed storytelling, diary studies, or interviews. The number of incidents needed depends on what is being studied, but generally a sample size of 50 to 100 incidents is enough [26, p.50].

There are five main identifiable events that can be collected with the help of CIT. CIT is useful to identify: the events that lead to the critical incident (the incident cause), the users behaviour during the incident occurrence (the user actions) what the user felt both during the incident and after the incident (the user sentiment), if the incident changed the user's behaviour and other potential outcomes if the behaviour was unaffected (the incident outcome), and what new potential outcomes that are possible in the future in the case where the incident does change the user's behaviour (the ideal outcome) [26, p.50].

The positive incidents and the negative incidents should be analyzed independently

from one another, and then reported in the same way. The data analysis is intended to summarize the findings into data summaries that can be used for implementations and conclusions regarding both the positive and the negative incidents. With this data it is possible for the team to prioritize recommendations [26, p.50].

4.10 Questionnaires

Interviews and questionnaires are the two primary tools for collection of survey information [26, p.140].

Multiple factors have to be considered when designing a questionnaire to ensure a good response rate. These are, amongst others, the appearance, clarity, instructions, arrangement, and layout. The response and the analysis of the responses are affected by how the questions are constructed. Hanington and Martin mentions as examples of this, open-ended questions which can be an opportunity for depth of the responses, in contrast a closed-ended question is easier to communicate and analyze numerically [29, p.178]. Hanington and Martin also states that asking the participants to either rank their choices or split a constant sum between the different options will result in a better indication of the users' preferences than a single checked response will. Likert scale[30] questions are recommended by Hanington and Martin to keep neutrality in questions without missing out on the strength indication in a response [29].

While it is possible to use questionnaires independently they are usually combined with other methods, as observations, to complement the data that can be missed when the participants write their responses. The observation can also be used to either verify or challenge the behaviours reported by the questionnaire participants [29].

4.11 Evaluative Research

Evaluation should be done iteratively, never only on the final product. Evaluation research is an iterative process used to improve a product based on feedback from potential users. This feedback can be gained through presenting concepts and prototypes. However, it is also possible to perform evaluation research on already existing products which has a potential to aid early stages of design research. Evaluation research can be both flexible and tightly controlled. If the fidelity of the prototypes is high enough they can be tested flexibly in the correct context and in this way represent realistic usage well. With a more strictly controlled approach the prototypes or products can be tested with control over other aspects that may influence the test, this may however represent reality worse. If exploratory and generative research has been conducted first, then the need of evaluation research can commonly be for verification, to evaluate how well the designers have followed the users' feedback [26, p.74].

4.12 Heuristic evaluation

A heuristic evaluation is done before testing with users. To perform a heuristic evaluation the team agrees on certain usability related criteria to assess the product against. When performed repeatedly it can help the team create principles to follow while designing instead of relying on one's intuition. It is recommended that multiple evaluators assess the interface independently before summarising all of the findings, to decrease the potential bias any one evaluator can have [26, p.98].

Heuristic evaluation can help identify missing elements but finding defining design opportunities through it is a rare. When performing the heuristic evaluation the identified problems, findings that do not align with the set up criteria, should be reported, preferably with images. Things that should not be changed are also common to include, together with images of those things. Reporting both strengths and weaknesses helps balance the report, and including the strengths can help motivate new iterations [26, p.98].

Performing heuristic evaluations can make future usability tests more efficient due to the opportunity to fix problems with the prototypes before including users [26, p.98].

4.13 Interviews

Interviews can be structured, unstructured [26, p.102], or semi-structured [31]. A structured interview is advantageous if control over questions and time is important. While an unstructured one is advantageous if flexibility is important, even though the unstructured option is more flexible the interviewer usually aims to address a set of predetermined topics [26, p.102]. A semi-structured interview is well suited when follow-up questions to initial open-ended questions are required. According to Adams semi-structured interviews are worth considering when the interviewer is researching an area with unknown issues and the interviewer therefore needs to be able to pursue any leads discovered during the interview [31, p.492-493].

The unstructured version works well for exploratory purposes whereas the structured one is more appropriate when consistency is important, in this case bias can be avoided by always reading the questions exactly as they are written [26, p.102].

It is common for interviews to be complemented with questionnaires or observations [26, p.102].

4.14 KJ Technique

When using the KJ technique each member of the team identifies what concerns they have and what they require from the project, individually, on sticky notes. Concerns and challenges that are of similar nature are clustered together. All of this is done in

silence, which could help with projects that suffer from misrepresentation as coercion is no longer possible [26, p.105].

4.15 Literature Reviews

Literature reviews can be useful in design projects. It is intended to gather important parts of existing research to guide the active project. The entirety of each source does not have to be summarized [26, p.112].

The selected literature should be from credible sources in addition to being relevant for the project. The material selected for the literature can be divided into categories depending on the research area the literature best correlates to [26, p.112].

4.16 Observation

A structured observation is appropriate if behavioral or environmental aspects are outlined well and in focus. Outlining these aspects is commonly done through prior pilot observations performed in a semistructured manner [26, p.120].

If the designer is unfamiliar with the area to be observed a semistructured observation can be suitable. For this it is possible for the observer to have questions to guide to observations but focus should be on having an open mind, it is also allowed to deviate from the plan. Although deviations from the plan are allowed, the observations should be systematic. It is also important to document the observations well, this can be done by taking notes, sketch, photograph, or record video. The information gathered is typically combined into a coherent whole to guide design inspiration [26, p.120].

When documenting observations the observers should differentiate between objective observations and subjective interpretations of behaviours. Conclusions can be confirmed by the participants via interviews, either during the observation or after it has concluded [26, p.120].

4.17 Parallel Prototyping

According to Hanington and Martin, iterative design has been criticised for sometimes fixating on an inferior design approach. Parallel prototyping, when performed before the iterative process, can help the team consider more extensive range of design opportunities. This is what parallel prototyping intends to help with, to enable a simultaneous consideration of many different design approaches before choosing and elaborating on a particular one [26, p.122].

When performing parallel prototyping the designers of the team should individually design multiple low-fidelity-prototypes, these designs should then be tested by end

users or assessed through an heuristic evaluation by experts. The intention of the evaluations is to help designers determine which aspects of the prototyped designs that induce the desired reactions from the users [26, p.122].

There are multiple advantages with parallel prototyping. As parallel prototyping aims to produce several designs it creates an option to get feedback on multiple design approaches, in this way it encourages consideration of various designs. It is easier for the designer to not take criticism personally when several designs are evaluated simultaneously, in this way the focus is shifted to the designs instead of the designer. It can help decrease the amount of rivalry within the designer team as it is common for parallel prototyping collaborators to incorporate parts of each others' concepts into future designs. Parallel prototyping is because of the aforementioned advantages an appropriate method for teams that can focus too heavily on a single approach early on [26, p.122].

4.18 Personas

It is important for the designers to understand the users in order to be able to create a user-centered design. Trying to satisfy everyone's personal preferences when designing will result in an incoherent outcome, because of this different people's preferences has to be combined into a coherent whole. Personas is a good solution for this as they should be created by collecting information from actual users [26, p.132].

When multiple users can be described with the collected information it is time to find shared features and attributes. The findings should be combined into a coherent description, a persona, this process can be done by affinity diagramming (described in section 4.4). To avoid a too broad design focus and traits that differ too much from other members of the group the number of personas should always be limited, three to five personas are enough [26, p.132].

The description of a persona is commonly one page, or shorter. It should include a a name for the persona, a stock photograph or sketch to represent the person, and a description of important aspects of the persona, such as aims, life situation and relevant behaviours. It is allowed to include subsidiary images of the persona's usual activities, locations, and objects for a more convincing impression [26, p.132].

Instead of actual humans the personas can now be used as a reference in every phase of the project. The personas can help lead development and discussions as well as presentations of products or system design. Not only is the persona beneficial for the design team, it can also help communicate with the client [26, p.132].

4.19 Stakeholder Maps

Stakeholder Maps are primarily usable for planning, scoping, and definition [26, p.5-7].

The designers identifies what the stakeholders consider key constituents. These constituents are drawn to show relations between each other. Stakeholder maps are usually generated through speculative Brainstorming at first, with people most involved in the project attending to specify the end users and anyone who will benefit from the project [26, p.166].

4.20 Stakeholder Walkthrough

Stakeholder Walkthroughs are usable for concept generation, and early prototype generation [26, p.5-7].

Stakeholder Walkthroughs are used during early prototyping to bring together the end users, the stakeholders, and the design team. This is in order to provide feedback to the stakeholders and design team while building empathy towards the end users [26, p.168]. All of the previously mentioned parties gather in a room while following the Think-aloud Protocol during an evaluation (see Section 4.22). It is important to consider the end user the *primary participant* and say this up front, as stakeholders may feel ignored otherwise [26, p.168].

4.21 Surveys

Surveys are mainly usable for exploration, synthesis and design implications [26, p.5-7].

Surveys can gather many different kinds of self reported information, such as thoughts, feelings, and behaviours. Surveys are commonly used to gather quantitative data from a large sample size in a relatively short amount of time [26, p.172].

The two most used techniques for collecting data via surveys are questionnaires (see Section 4.10), and structured interviews (see Section 4.13). Questionnaires are either sent to participants for them to complete, or they are read to the participants by the researcher. If these questionnaires are sent to a large amount of people, it is important with carefully worded questions so it is possible to analyze the results of questionnaires statistically [26, p.172]. Interviews are described in greater detail in Section 4.13.

4.22 Think-aloud Protocol

The Think-aloud Protocol method is usable for concept generation and early prototype iteration, as well as evaluation, refinement, and production [26, p.5-7].

During the Think-aloud Protocol, participants are expected to verbalize each step of a *task* that they are currently performing, which could answer questions such as *what* they are thinking, doing or feeling in that moment. This may reveal positive or negative aspects of the interface [26, p.180].

There are two main ways of conducting the method, Concurrent and Retrospective. During the Concurrent Think-aloud it is entirely allowed and recommended to remind the participant to verbalize themselves during the Think-aloud Protocol in order to get as much relevant information as possible. During Retrospective Think-aloud the designers and participant is quiet until after the task is completed, the entire activity is recorded in some way and the participant is then allowed to re-watch her actions and comment on them [26, p.180].

4.23 Triangulation

The Triangulation method is usable for exploration, synthesis, and design implications, and also for concept generation and early prototype iteration, as well as evaluation, refinement, and production [26, p.5-7].

The goal of Triangulation is to compare different results against each other to see if they are essentially the same. Getting several answers that answer the same question will result in more confidence in future design choices. Commonly when answers need to be triangulated, the answers were given though different means (such as both questionnaires and interviews). Triangulation could be effective when comparing qualitative and quantitative data collections against each other to more quickly realize where the answers support or contradict each other [26, p.188].

4.24 Usability Testing

The Usability Testing method is usable for concept generation and early prototype iteration, as well as evaluation, refinement, and production, and also for launching and monitoring [26, p.5-7].

Usability Testing is used by designers to evaluate a digital application, as an individual experiences it through a set of different tasks. This testing is done in order to figure out which parts of the interface that is confusing, so they can be fixed before an actual launch. These tasks are usually end-user goals that reflect the goals of the target audience, and should be as clear and specific as possible [26, p.194].

Usability testing often reveals in which ways the typical end user interacts with the

system in a different way than the developer or designer. Typically, as you run usability testing through more people, even more problems will be revealed with each test [26, p.194].

4.25 Weighted Matrix

The Weighted Matrix method is useful for concept generation and early prototype iteration [26, p.5-7].

This method is particularly useful when the team has generated a lot of ideas in the early stages of the process, and the team needs to decide. It generally helps with gaining a shared decision making process, and helps to remove the biases towards certain ideas [26, p.202].

When executing this method, the designer weighs different areas against each other. The more important the area is, the more or less weight it gains in comparison to the other areas. The most common items to list is *opportunities*, in order to rate the future product success on a scale. The result is still subjective and qualitative, but makes it possible to rate the product based on success rather than opinion [26, p.202].

4.26 Wizard of Oz

The Wizard of Oz method is usable for concept generation and early prototype iteration, as well as evaluation, refinement, and production [26, p.5-7].

The Wizard of Oz technique tricks the unknown user into believing that they are interacting with a fully functioning prototype, but in reality is partly or fully handled by an unseen researcher. The participant's path through the system is shaped by the researcher who now can evaluate the system without it needing to be fully functioning [26, p.204].

The setup requires the researcher and the participant to be in two different locations. It therefor helps to have the opportunity to observe, either through video or screen-sharing. In the early stages of prototyping it is probable that *all* of the actions must be simulated by the researcher, but in later stages of testing the researcher should be able to intervene less often [26, p.204].

5

Planning

In this chapter we present the goal of this project as well as how we plan to reach that goal by making use of various methods.

5.1 Scope

The goal of this project was to provide Kollmorgen with design prototypes for the Layout Designer software, with related documentation to aid further development of the product. The focus was to improve the user experience and learnability for new users, while trying to not negatively impact the productivity of experienced users. While it is hard to measure decreased productivity for the expert users, it should still be held into consideration when making the design decisions. A very relevant quote from *About Face* regarding efficiency says that "A general guideline of interaction design that seems to apply particularly well to productivity tools is that good design makes users more effective"[7, p.16]. In order to provide knowledge to others reading this report, a set of guidelines was planned to be included for anyone seeking to develop similar software.

It was not within the scope to develop a functioning software product for Kollmorgen.

5.2 Process

The thesis work was planned to start of with a prestudy with focus on graphical user interfaces and usability. This was to confidently be able to identify problematic areas of the old software and apply improvements where needed.

We estimated that approximately two days a week would be spent writing on the report, that being the overall average.

In order to get a better understanding of the program that was going to be evaluated we planned to complete the existing tutorial provided by the company.

To reach our desired goal, design prototypes were planned to be evaluated. To im-

prove our interface prototypes, we wanted to familiarize ourselves with Photoshop and Illustrator, we also planned to use Figma, an online user interface design tool [32], for prototyping but we were experienced enough with that to exclude it from the familiarization phase. We planned to do preparations for interviews (section 4.13), questionnaires (section 4.10), and observations (section 4.16) in parallel to our familiarization phase. The inquiries were planned to be performed in the ensuing weeks. We wanted to perform all three types of inquiries because they can complement each other as they are good for gathering different types of insights.

The first interview was planned to be with subject matter experts, mentioned in section 4.3, since experts can provide insights which could not be found by ourselves in the given time. We planned to summarize all findings from the aforementioned inquiries, and if we could get enough data also create a persona to use in future design discussions. However, we did fear that the amount of data would be insufficient to create a reliable persona and planned to use Affinity Diagramming (section 4.4), as a backup method to summarize the data and get the best available reference for future design prototypes. As mentioned in section 4.18, Personas is a good resource when trying to create a coherent design, this is the main reason that we wanted to use the method.

The current software was planned to be evaluated with new users, and make use of the think-aloud protocol from section 4.22 to grasp the users' expectations and feelings, to be able to adapt our design more closely to the users' mental model (section 3.2.4).

As usability is of high importance, both to reach a conclusion regarding the research question and to satisfy Kollmorgen's wishes of a less steep learning curve for new users, we of course planned to perform usability testing, discussed in section 4.24. To not get stuck on a single design approach we planned to employ the usage of parallel prototyping (section 4.17) early on in the design process.

The process was planned to be iterative to continuously improve on the latest produced prototype of the program's interface. We were expecting to be part of an employee team at Kollmorgen which works in sprints, which naturally means we would, to an extent, work in sprints as well, to be somewhat coherent with the workflow of the team.

Dates for observations and questionnaire send outs (visualized in Figure 5.2) had not been confirmed and were likely to change but we hoped to be able to execute them as early as possible, to build an understanding of how Layout Designer is used and be able to accommodate that with our proposed designs.

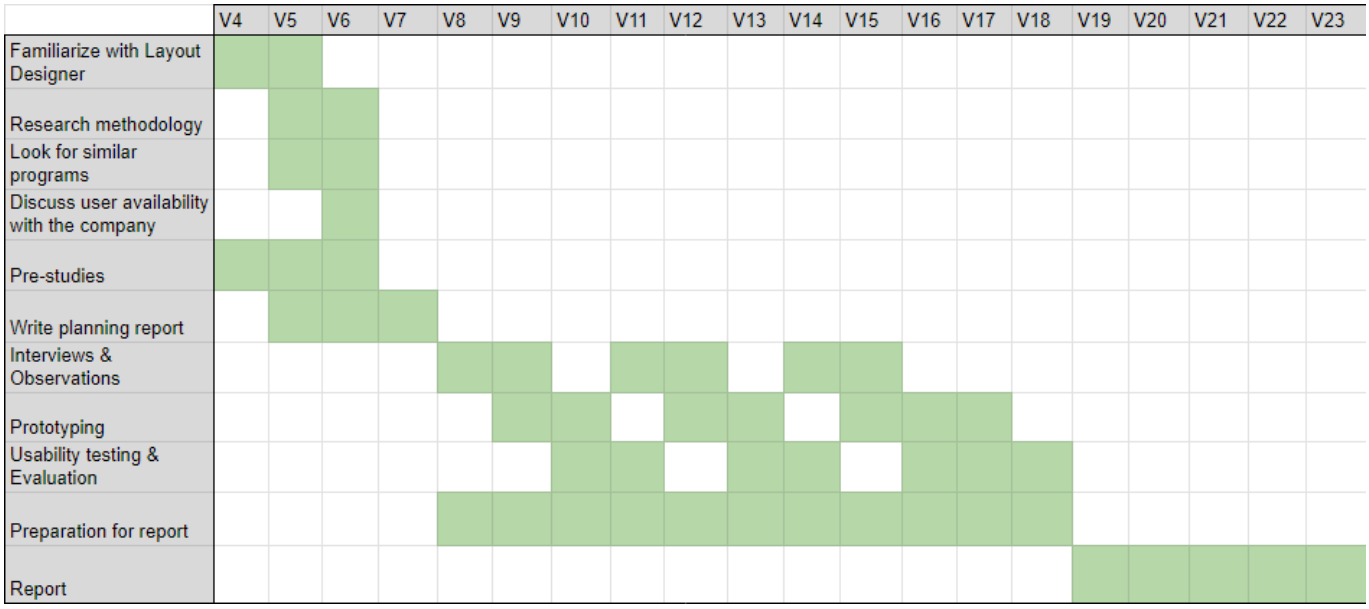


Figure 5.1: A Gannt Chart with the estimated time frame from before the prestudy was performed

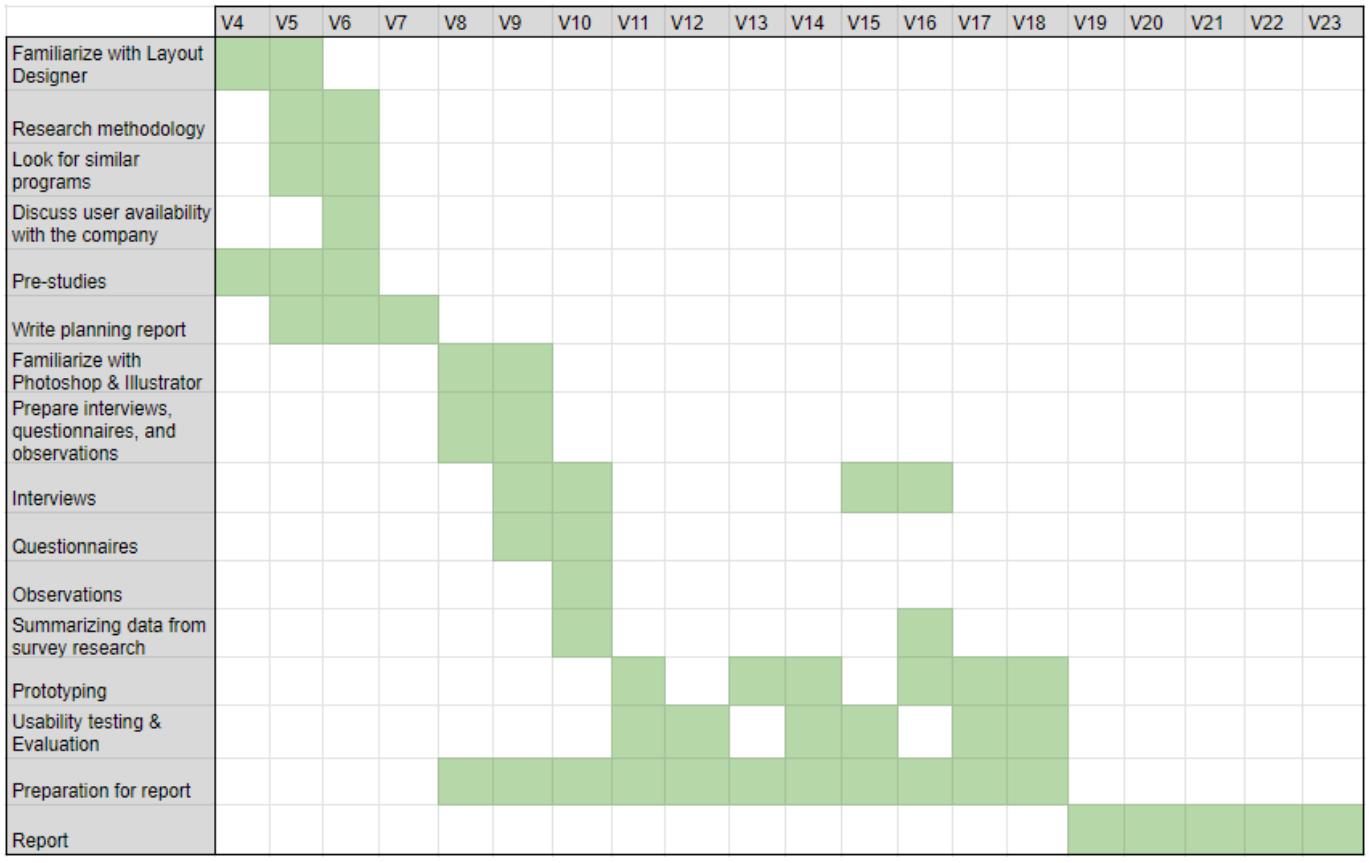


Figure 5.2: The new Gannt Chart representing the time plan for this project

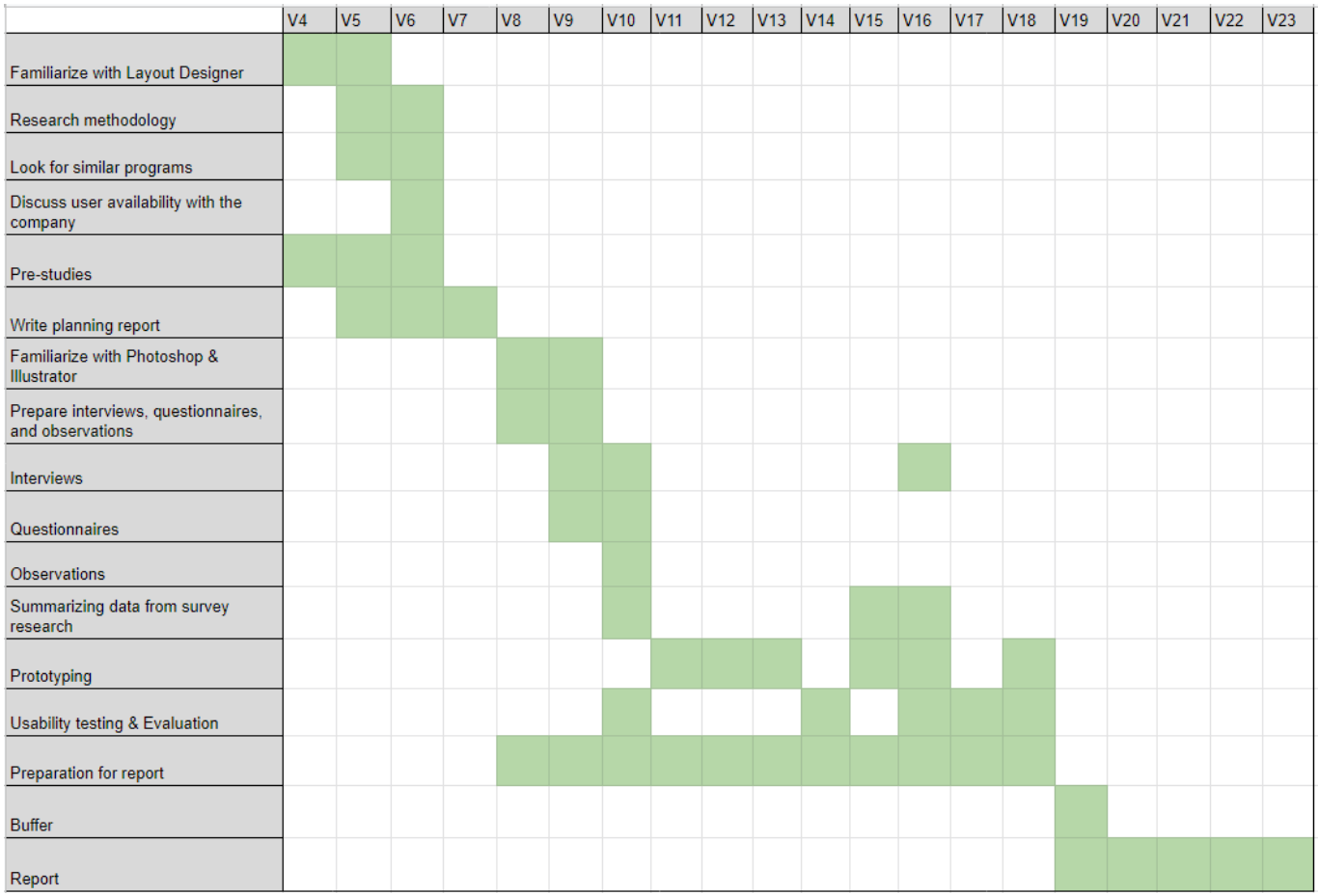


Figure 5.3: A new chart that is updated due to the Corona Virus

5.3 Time Schedule

6

Execution and Process

During the course of the project, we have tried to follow our previously mentioned time plan, and will now discuss what actually happened.

6.1 Prestudies

During the time that we explored the application by ourselves, we detected several problems that correlated to severe and rather obvious design flaws that were taught to us during the course of our master's programme. Some examples of these flaws will be given in the following subsections.

6.1.1 Analysis of current software

During this analysis, the main (or at least most obvious) problematic areas of the software is discussed. For further examples of problematic interactions and potential solutions, we refer you to Appendix B.

6.1.1.1 Badly Timed Feedback

In section 3.2.1, information is given regarding how software some times may give unnecessary information to the user. To give an example of an error message where it is not only hard to understand what went wrong from the descriptive text, but also where the error message appears several steps after the incorrect input. Figure 6.1 shows how there are several input fields needed to continue to the next step, if any of these fields are empty, a modal popup is shown that tells the user to fill them in to be allowed to progress. After pressing "next" two times and then confirming, the user will be met with Figure 6.2 in case the "Drawing Name" was not a file path to a CAD-drawing. When pressing OK, the process made by the user this far is discarded and has to be done a second time.

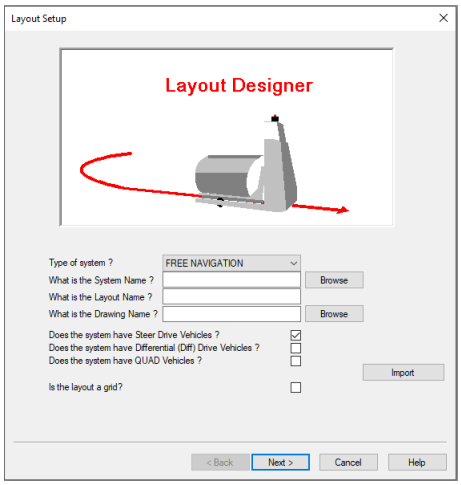


Figure 6.1: When creating a new layout in Layout Designer, this is the first out of three obligatory setup-steps.

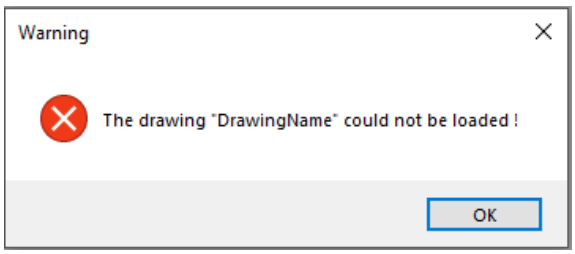


Figure 6.2: The error message shown when not including a correct Drawing Name.

6.1.1.2 Disrupting Workflow

One of the more serious issues that Layout Designer has, is regarding to workflow. The most apparent problems involve path drawing errors (Figure 6.3, 6.4, 6.5), and the tool placements in the toolbar (Figure 6.6).

Errors that are similar to Figure 6.2 can be found in more occasions than in the creation of the Layout, and can frequently be found when drawing driving routes. When drawing a path, there are warnings that can occur that will remove all of the progress made on the route being drawn. The three main reasons that this may happen, that we have found, are:

- The drawn path is longer than 40 points, as shown in Figure 6.3.
- Two of the points are too close to each other, as shown in Figure 6.4.
- One of the points are outside of the drawing, as shown in Figure 6.5.

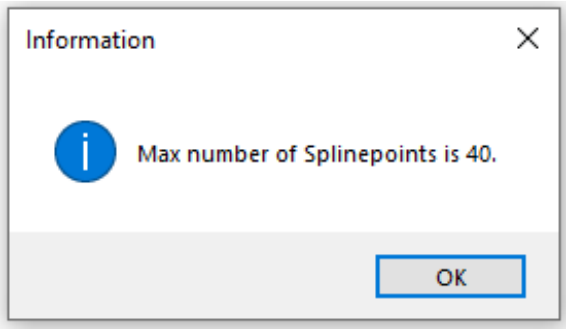


Figure 6.3: When the 40:th splinepoint has been placed, this modal popup shows up and removes all of the splinepoints.

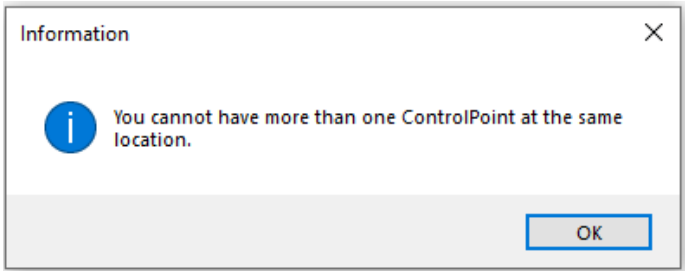


Figure 6.4: When drawing routes, this error shows up if two points are at the same location. The error pops up as soon as the second point is placed and removes all progress on the current route.

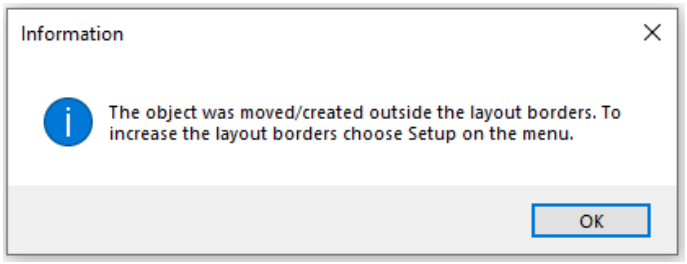


Figure 6.5: This is the message shown when something is dragged or created outside of the layout border.

A special case regarding the error in Figure 6.5 is when Layout Designer automatically optimizes the route and partly appears outside of the screen. Then it is impossible to change it back without having to delete the path first.

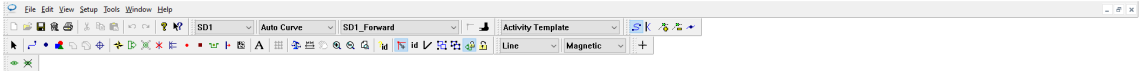


Figure 6.6: This is the *default* toolbar layout in layout Designer.

As can be spotted in the toolbar in Figure 6.6 (or Appendix C.7 for a larger Figure), there are different segments that can be moved around as you see fit. There are two different tools that are enabled from the start, namely the *Snap* and *Reflector Lock*. Since certain tools are located on the same segment in the toolbar they can not be separated, this makes the distance between them uncontrollable for the user. Certain items on the toolbar are selected, while others are activated, and there is no specific way for the user to realize which is which other than learning it through using the application.

6.1.1.3 How Layout Designer fails to meet user goals

The characteristics of a program failing to meet user goals are written in section 3.2.3. This chapter intersects partly with the information in section 6.1.1.4, since being presented with a lot of extra information or extra steps in order to complete a task ultimately makes the user feel stupid and requires too much effort to operate effectively, which is two of the main reasons that an application fails to meet user goals.

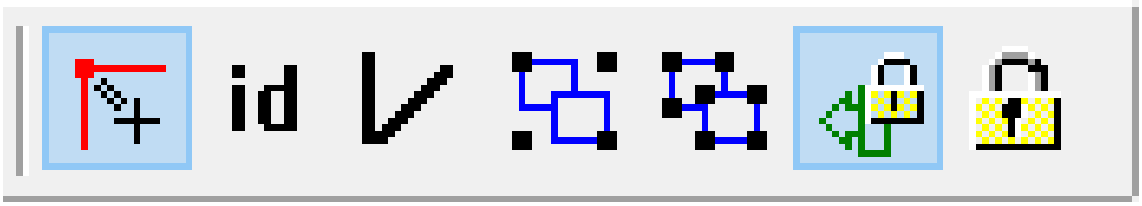


Figure 6.7: A zoomed in view on part of the toolbar with the *Snap tool* and *Reflector lock tool* enabled.

When first being introduced to the program, it is not very unlikely that the user will try to use most of the tools and eventually attempt to drag them around. While the *Snap tool* is enabled, paths and walls can only be dragged in a horizontal path to what they face. While the *Reflector lock* tool is enabled, the user will not be able to move or rotate any reflector. The impression that each line or reflector needs to be perfectly placed from the beginning could be discouraging, and when the user realizes that this is not the case, it will undoubtedly make them feel stupid. If the Snap tool or the Reflector Lock tool is disabled by the user, they will be re-enabled once the application restarts.

6.1.1.4 Mental Model of Layout Designer

In regards to how well Layout Designer's represented model (explained in section 3.2.4) manages to reflect the user's vision, is hard to say for sure before any official interview has been conducted.

When creating a path using the Spline Tool, a popup is displayed with information regarding which optimizations were done (as can be seen in Figure 6.8). Here the user is met with information and numbers that they are not particularly interested

in. Why does it matter to the user that the number of iterations done is 69? In this case, Layout Designer is obviously very close to the implementation model (explained in section 3.2.4), as it displays this information without any visual queues for what has actually happened.

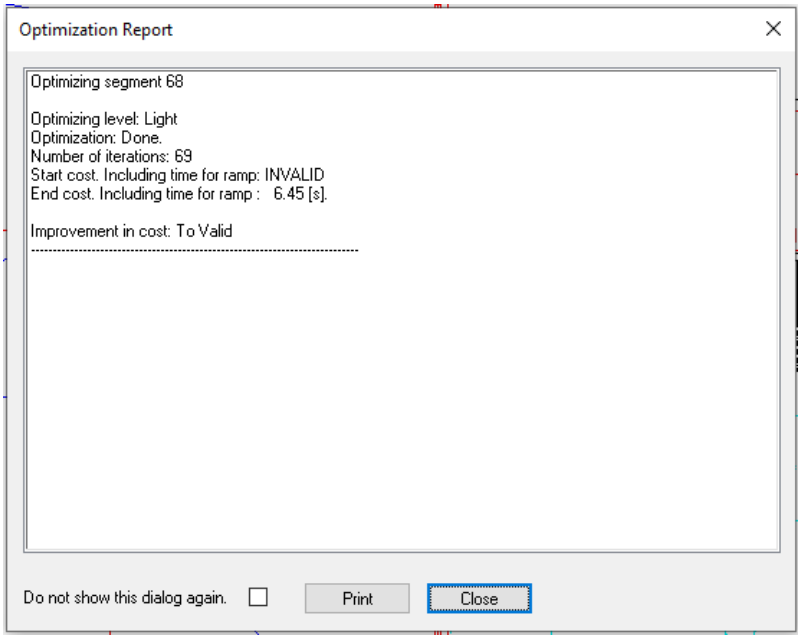
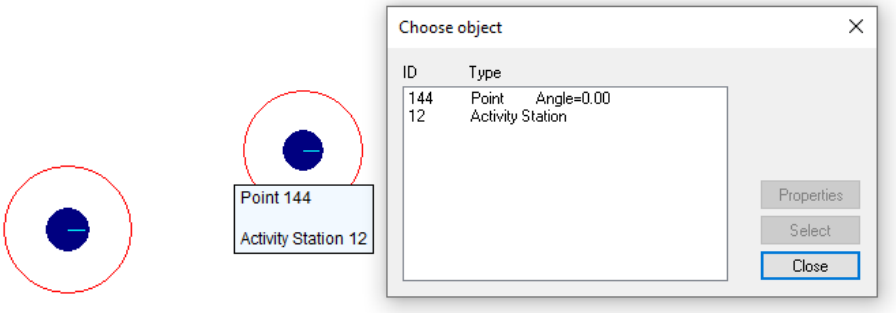


Figure 6.8: This is the popup that is shown when Layout Designer optimizes a spline-curve.



(a) **(b)**

Figure 6.9: Figure 6.9a shows how a station looks when it has been placed. When the blue area (the point) in the station is pressed the popup in figure 6.9b is displayed which allows you to chose which of the the objects you want to select.

6.1.2 SME Interviews

The subject matter experts were three employees at Kollmorgen, they all had extensive experience with Layout Designer. Two of them had in some way been responsible to teach others to use the software.

6.1.2.1 Preparation for Interviews

The interviews were prepared for each of the different groups, as we wanted to tap into their general experience of what makes them SME. We prepared the interviews to be semi-structured which generally proved to be a good choice, this is because we were shown a lot more than anticipated in the walk-through's that were provided. Certain intervals of time that each of the questions or tasks should take was added to the template, but this was followed relatively freely as we evaluated if the information we were getting was worth it in real time.

We were completely prepared to change some of the questions in order to ultimately provide us with better feedback from future interviews, but luckily pretty much all of the questions provided valuable and unique information. Before the interviews, each of the participants were asked if they consented to recording of their voices and the screen (which was done with the program *OBS: Open Broadcaster Software*[33]).

6.1.2.2 Execution of Interviews

All of the interviews were done in meeting rooms, as they do not differ from the usual working environment too much and also allows the usability tester to focus on the tasks. The interviews were combined with recording of both voice and screen, at the same time as both of the researchers (us) observed the screen on which Layout Designer was prepared with a general layout. The recording was generally done in order for both of us to be concentrated on the interview rather than writing down information, and made it possible to go back to the recording to get exact quotes in case we needed it.

The last SME interview was performed via Skype and was combined with screen sharing. During this interview the sound was unfortunately not recorded. We had taken some notes, and with help of these we started watching the screen recording to remember relevant information the interviewee shared with us. A picture of the interview can be seen in Figure 6.10.

6.1.2.3 Affinity diagrams

As previously mentioned, *Cooper et al.* primarily suggests that making personas is a great way to summarize the data provided by the interviews and other data gathering methods [Chapter 4.18]. But since we only had 3 people that were available for interviews, creating personas would *not* be feasible as we would not have a fair representational overview of the users. We decided that the affinity diagram would still be usable for us, even though we did not use the information to create personas.

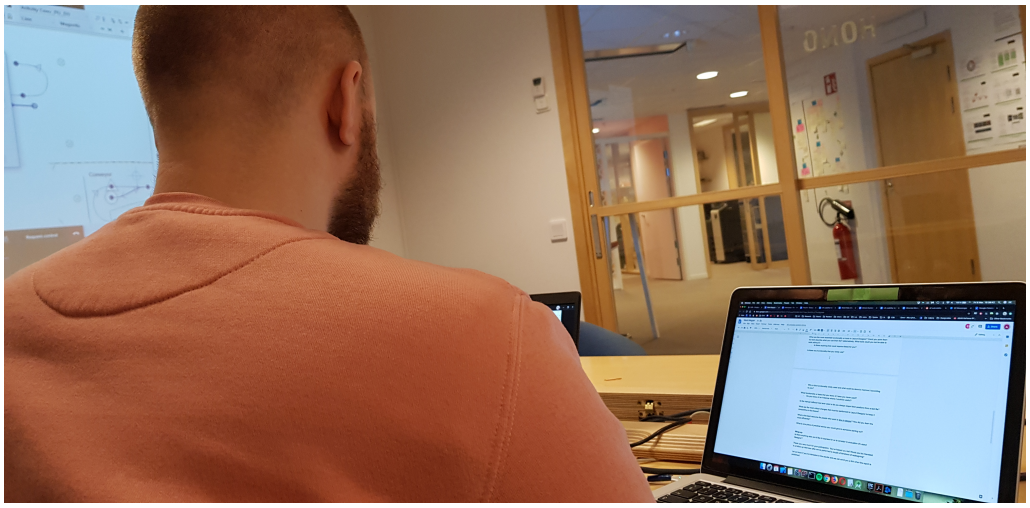


Figure 6.10: A picture from our SME Skype interview where the actions of the interviewee were showcased on the projector, which was recorded, while we took combined notes on our two computers.

(a) **(b)** **(c)** **(d)** **(e)**

Figure 6.11: Our affinity diagram process. Figure (a) to (c) represents the first version, while (d) and (e) represents the second version.

We created several different affinity diagrams, partly because the interviews were spread out but also because we created the categories in a premature state. We will still describe the process of each of the iterations, but it is not advised to follow our first version as it does not correctly follow the affinity diagram process.

Our first version of the affinity diagram was done after the second SME interview, but before the third, since the interviews were roughly two weeks apart. This version was done incorrectly and should probably not be considered an affinity diagram, but rather a categorization (which proved useful in future versions). As the interviews were recorded we used these recordings to write down notes from. Everything we believed was noteworthy was written down on a post-it, as this was one of our first thorough showcase of the program a majority of the what was said during the conversations was written down.

During our first iteration we created three different categories, namely **Suggestions**, **Problems**, and **Descriptions**. In this stage the description category was a lot larger than the other two as can be seen in Figure 6.11b. The descriptions then got divided into three separate categories, which were **Informative Description**, **Problematic Description**, and **Spline Description**. We noticed that a lot of the information in the Description and Problems would fit well into two new categories: **Workflow** & **Experience**. Where workflow was any part that is related to maximizing the efficiency, and experience was either how long people have worked or information that is gained through experience.

This version served us in the sense that we got a more insightful view into what kinds of answers we had received from our first two interviews. After this categorization, we had our third and final interview which prompted us to start doing a more correct affinity diagram with the new data gained.

When the third interview was over, it was decided that the current categorization needed an overhaul. This was in order to not make the current categories too large with the newly found information, while having a chance to create the diagram in a correct way according to the Affinity Diagram guidelines.

We started off by picking the most obvious post-it notes and placing them together (such as experience within the working field), and then picked one post-it note at a time. In the end we had some general topics which we ultimately tried to name depending on the common issue within the group. Some groups had no issue, but were rather statements, and got a green post-it note to mark their name. The groups that were related to issues were given a blue post-it note (See Figure 6.11e for reference).

The topics related to issues that were gained from the final version of our Affinity Diagram were:

- Unexpected software limitations
- CAD-layer (issues)

- Lack of common component library
- Lack of functionality in check-function
- Mirror tool functionality issues
- Suboptimal toolbar
- Missing functionality
- Non-apparent settings
- Inconsistent movement restrictions
- Hard to distinguish individual elements
- Previous Layout Designer changes
- Prolonged learning curve
- Unclear safety fields
- Stations
- Point-point problems
- Point-segment problems
- Sweeps/blockings
- Station operation usability
- Unnecessary functionality
- Useful functionality outside of application
- Limited knowledge related to the tool
- Potentially disastrous mistakes
- File management

6.1.3 Novice User Interviews

At this stage we knew certain functionality of the application that was commonly used, this was learned from the previous SME-interviews and individual performing of the tutorials. We took several of the core tasks, and told users with no previous interactions with this program to perform these tasks with as little information as possible. These interviews gave more insight into which tasks were easy, hard, or impossible to comprehend without any prior tutorials done. The hard or impossible tasks were prioritized in the upcoming iterations.

We got a total of five participants for the tests of Layout Designer. To have a somewhat relevant user group for the tests we contacted Chalmers' Interaction Design and Technologies students, four were available to attend our test. The fifth participant attended the Systems, Control and Mechatronics programme. The main reason we chose to reach out to students at Chalmers was because they are all potential employees of Kollmorgen, and in that sense, potential new users of Layout Designer.

The interviews were conducted in a way that made the interviewees go through all of the tasks, which resulted in a discussion that was semi-structured to allow for additional thoughts and questions. The participants were asked to think out loud while performing the tasks. We recorded the screen to be able to observe all the tasks later, the conversations were recorded as well.

After all tests we watched the recordings and took notes of interesting aspects of

each interview. All notes were written in a Google Spreadsheet, in chronological order, to easily be able to see what different participants had in common during each task.

Multiple issues were identified through these tests. All participants had some trouble with the initial set up of a new layout. The software does not warn users of incorrect inputs until several steps later, forcing the user to redo everything even if they just had a minor error in the first step. The setup was expressed to not be intuitive and one participant even expressed feelings of being tricked. From the SME interviews we know that the spline tool is suggested when drawing paths, rather than the segment tool, but during our tests all participants picked the segment tool at first. Once the participants started using the spline tool they either expressed that it was easier to use or we observed an increase in efficiency while working with the splines compared to the segments. However, the participants were either confused, annoyed, or both, by the optimizations the software automatically performs when using the spline tool. It was also quite apparent that the participants had trouble using the different types of stations, everyone found the station tool in a reasonable amount of time, but changing the station template took long. At one point it was not found without assistance. The participants expressed that the position and states were confusing. The drop down menu is always active, so there is no indication that it only affects the station tool, and the default position of the drop down menu is placed far away from the station tool, as can be seen in Figure 6.12 (or Appendix C.8 for a larger Figure).

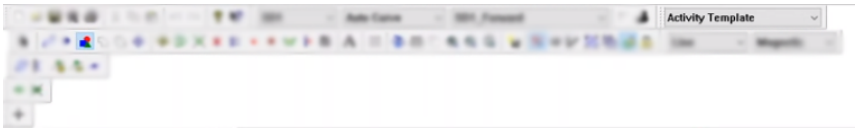


Figure 6.12: Selected station tool and corresponding drop down menu in focus.

6.2 First Iteration

During the first iteration multiple design approaches were tested. Efforts were focused on designing potential improvements for the most prominent issues found in the prestudies.

6.2.1 Prototyping

During the first prototyping iteration we tried to focus the most commonly expressed concerns new users had while testing Layout Designer while at the same time considering what the subject matter experts had told us about the software. To get a foundation to start from we chose to look at software we were fond of as inspiration.

6.2.1.1 Sketches

The paper prototypes both involve the feedback from the SME interviews, as well as the novice user interviews. These paper prototypes were used to allow us to

prototype in parallel to each other (see Section 4.17) to test the ideas that we both were uncertain of.

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 6.13: Some of the paper sketches made to accommodate the feedback. Figure (a) to (e) show general ideas and layouts, while Figure (f) focuses on icons.

We graded the sketched ideas together, while discussing the potential improvements and problems that may arise, and developed a grading scale that seemed appropriate. Ideas that were good, and needed to be implemented, were marked with a green marker. Ideas that were moderately good, or needed to be tested further in order to come to a conclusion, were marked with a yellow marker. Ideas that were bad, or generally not worth the time to explore, were marked with a red marker.

6.2.1.2 Inspired by other software

A very common feedback that was given by the novice users were that the current toolbar of Layout Designer lacks intuitive functionality grouping, and part of the feedback we got from the SME-interviews mentioned non-apparent settings, CAD-layer issues, and how the current file management system made it very difficult to merge between computers. This feedback combined with our own observations , were the main reasons for us to design this prototype, inspired by Unity and Photoshop, two software we believe handle some of these issues well.

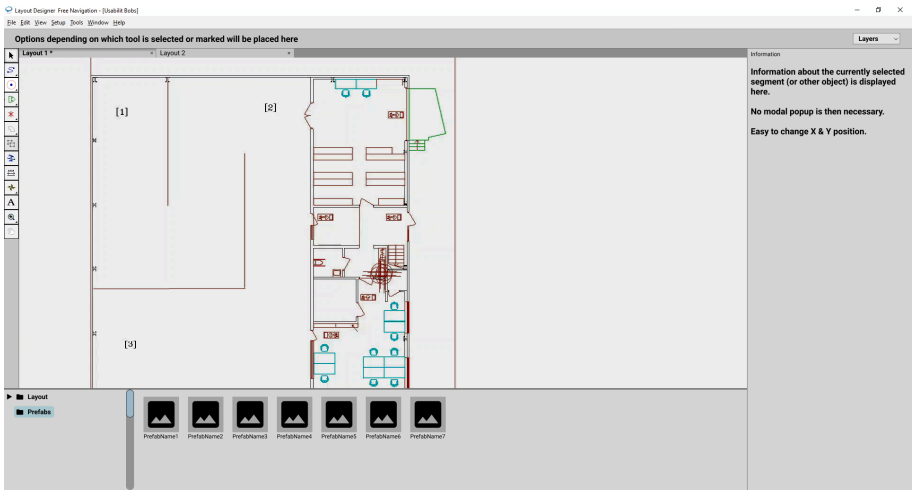


Figure 6.14: This design uses the toolbar design of Adobe Photoshop and the default screen layout of Unity.

6.2.1.3 Ideas to accommodate expressed concerns

One of the subject matter experts told us that he usually moves his systems and layouts from the folder that Layout Designer uses as he has so many that it becomes hard to navigate. This gave us the idea to improve the current functionality, a pop up with all files sorted by the system's name, shown in Figure 6.15.

A tutorial for the software exists, and it was mentioned multiple times during out SME interviews, but it is not part of the software itself, it can be found on a separate website. Based on this we thought it would be good to have a starting screen where it can be easier to navigate amongst the created layouts as well as have easier access to relevant tutorials. Our idea of this is shown in Figure 6.16.

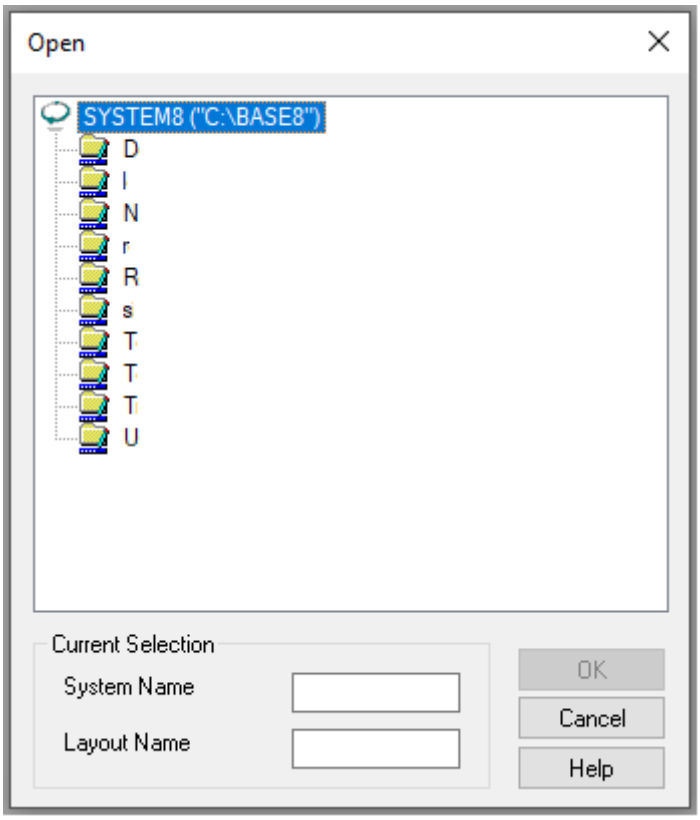


Figure 6.15: Layout Designer's window for opening a file

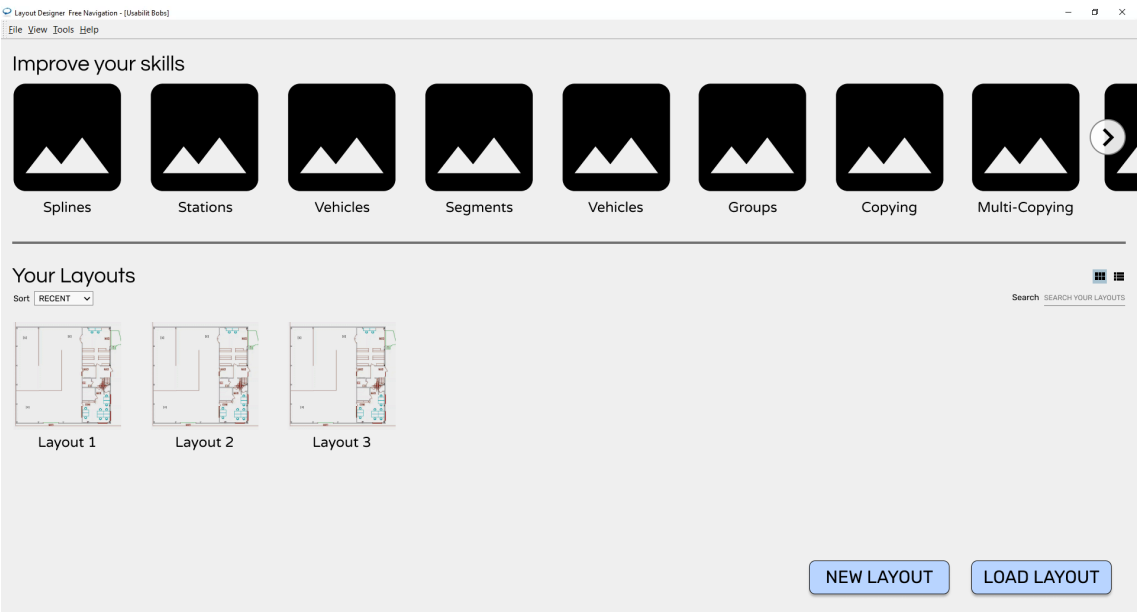


Figure 6.16: Layout Designer's window for opening a file

As mentioned in section 6.1.3 the participants found the toolbar to be counter intuitive. We had two ideas of how to accommodate this, similar tools can be grouped together in a better way, our approach can be seen in Figure 6.17, and

context sensitive functionality should only be presented to the user when it can actually be used, showed in Figure 6.18.

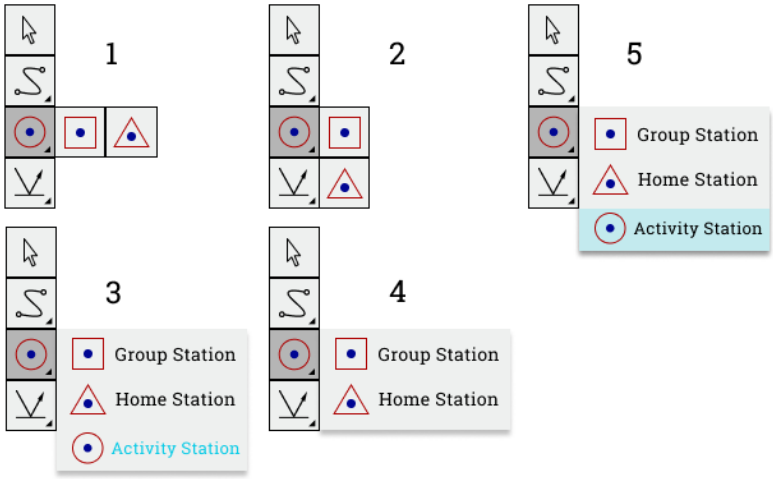


Figure 6.17: Ideas for how to show a group of similar tools

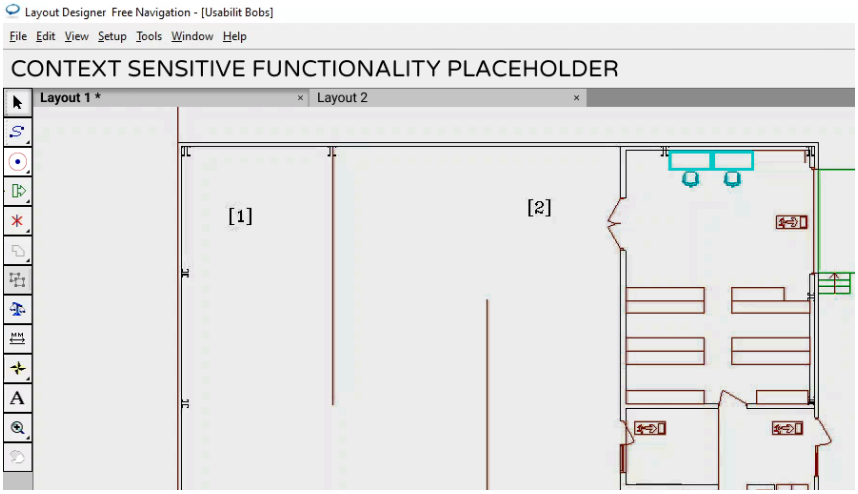


Figure 6.18: Idea for placement of context sensitive functionality

In the interviews with the subject matter experts it was brought to our attention that certain objects are sometimes copied between two layouts. To do this both layouts need to be open simultaneously, in two separate instances of the software. One idea to make this process more smooth for the user is to allow for multiple layouts being open in the same instance of the software, for easy navigation between the layouts we chose to place them in tabs in the main window, as can be seen in Figure 6.18. It was also mentioned that some objects could be created once and then used in a majority of future layouts. This also had to be done by copying between two open layouts. Our idea to decrease the excise was to give the option to save objects permanently that can later be accessed via all layouts. This would allow for easier usage prefabricated objects, nicknamed prefabs, and shown in Figure 6.14.

6.2.1.4 Observation based ideas

Although not much was said about how the selection was indicated in Layout Designer we observed during our tests with the novice users that sometimes they would miss what was selected. Due to this we chose to design a few alternatives to test in a future usability test. The alternatives we designed can be seen in Figure 6.19.

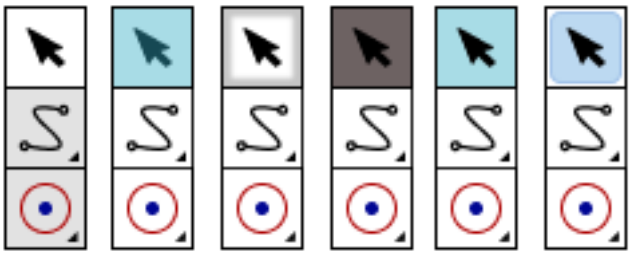


Figure 6.19: Different ideas for visualization of selection, illustrated here as selection in a partial toolbar.

While not explicitly expressed during the SME interviews that the way the software handles layers is inefficient, we did observe how it could take multiple attempts to interact with the correct layer in the CAD background. This was especially evident when the CAD background layers did not have specific names. Due to this we also chose to explore some alternative ways to visualize the layers and how to interact with them.

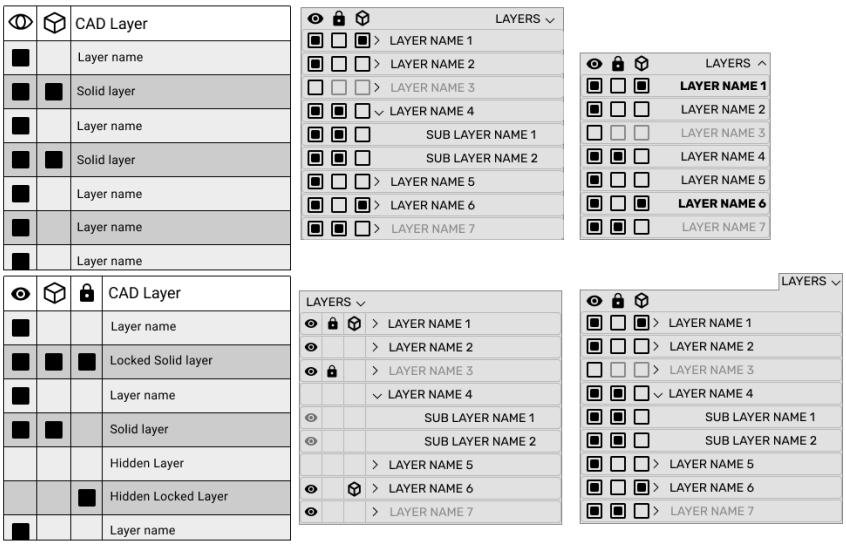


Figure 6.20: Options for layer management.

We also observed that a majority of all settings for different objects, some used frequently, had to be changed via a dialog menu. In order to decrease the amount of excise we believed it was important to decrease the need of dialog windows. Our idea was to move these settings into a permanent sidebar. This had multiple reasons, of

course the most important one was to supply access to relevant settings without the use of a dialog, but also to provide the user with more easily accessible information about which settings that are available for the currently selected object as well as not blocking the view while changing the settings that have a visual impact. The sidebar is shown in Figure 6.21.

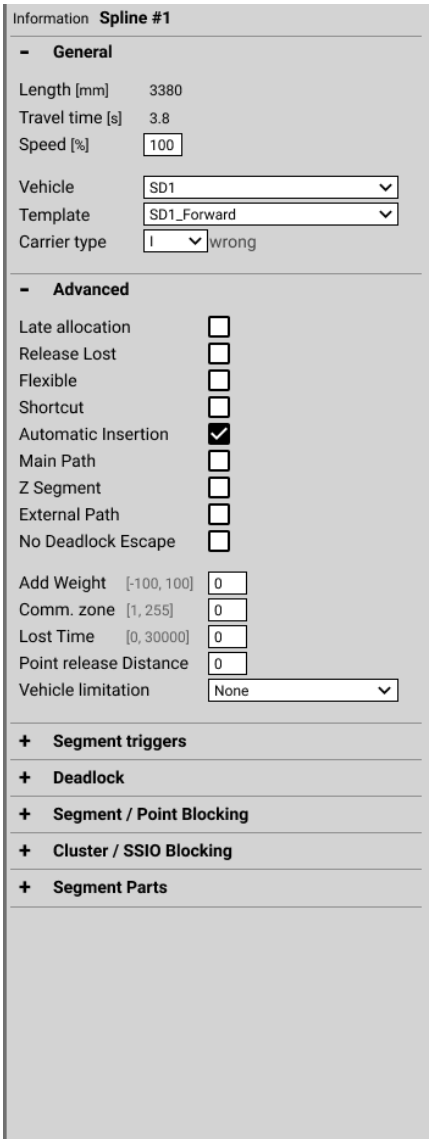


Figure 6.21: Settings in a sidebar

6.2.2 Evaluation of designs

For this iteration we wanted to get input from more people than before, so we used a questionnaire to get quantitative data about our current ideas, however we only got 26 people to respond to the questionnaire. We also wanted to make sure that our design alternatives were going to converge into a design which aligns well with current users of Layout Designer. Our original plan for this was to observe a few users in their natural work environment. Unfortunately this was cancelled as we did

not manage to book anything with users in Gothenburg, so the alternative was for us to travel to Stockholm, which we chose not to do as COVID-19 had recently begun spreading in Stockholm. Instead of these observations we then tried to retrieve some relevant information through questionnaires.

To get some qualitative data as well, we performed a formative evaluation with three participants, two graduated interaction designers and one web developer.

6.2.2.1 Questionnaires

For our questionnaire with quantitative data, we did not get many responses at all when the questions were a little longer and marked as required. This made us change the questionnaire to one with less required questions and easier questions. This change increased the responses from 3 to 26.

The observations that changed into questionnaires were not successful, we spent time on more thorough questions that we could ask in person. First it took a long time until we got any information on when and where a potential visit for our observations could take place. As mentioned earlier, COVID-19 had started spreading in Sweden, so we chose to attempt to gather similar information via questionnaires. The reason we chose questionnaires instead of interviews were because of our experience with how long we could potentially have to wait until those could be executed, especially considering we had already waited for the information regarding the visit. Based on our experience with the aforementioned questionnaires we chose to do a short questionnaire where not all questions were required. We asked for contact information to be able to send our questionnaire to Layout Designer users but only got four email addresses. Even though we designed the questionnaire to be quick we only got three answers, even after sending a reminder, so we believe that this is not a significant enough amount of data to base any decisions on.

6.2.2.2 Formative Evaluation

This evaluation was done to make sure our ideas were actually considered to be improvements by people with relevent technical experience. As mentioned before (see Section 6.2.2), two graduated interaction designers, and one web developer, were asked to give feedback on our ideas in comparison to the equivalent part of the old software. We asked if they thought specific things were improvements, which alternatives they liked, if they would change anything and in that case what. It was similar to the questionnaires but with focus on qualitative data instead of quantitative data.

6.2.3 Guidelines

As with the prototypes we intended to iterate the guidelines to get a better result in the end. At this point the guidelines had to be based almost entirely on literature as we were still in the beginning of the project.

Cooper *et al.* mentions "ignorance about real users" as one of the four main reasons to why digital products fail. The other three being misplaced priorities, conflicts of interest, and lack of a design process [7, p.6].

To avoid conflicting interests our belief at this point is that it is important to set up a goal that the designers as well as other stakeholders are comfortable and satisfied with. When a goal has been set, it is important to create a plan for the project, we believed this can be helpful in order to mitigate misplaced priorities. To prevent ignorance about real users it is important to include real users in the design process, to do this correctly it is important how to design for people. We suggest a structure mentioned by Tidwell *et al.*[3, p.2], find out who is part of your audience, what their goals are, and their behavior in regards to interface design. Personally we did not think that lack of a design process was something we had, we had planned interviews and prototyping, but we did not plan properly how to prototype. When we started to make prototypes we quickly realised that we had slightly different visions for the redesign of the software. Therefore we started over, started with sketches to make sure we agreed on some alternatives that we could then make higher fidelity versions of in Figma [32] and Adobe Illustrator [34].

We performed SME (subject matter expert) interviews, which we found valuable. If performing a redesign of an already existing software, we would recommend these, but that type of interview is not applicable when designing a completely new product as there are no experts of that software.

As previously mentioned, considering the users is important. We noticed that participants for interviews and observations can be hard to schedule. Interviews were delayed a little from our initial plan and observations were abandoned due to the COVID-19 virus and only being available in Stockholm. We suggest that these aspects of the project, that heavily rely on third parties, are planned early on, not only on the designers part but confirming dates with other participants as well. In an attempt to compensate for the missed observations we sent out questionnaires to get data from those users. It took some time before we were given email addresses to the Layout Design users, this means that even questionnaires might need to have a big time frame from initiation to finish if it is being sent out to a specific user group.

The guidelines we had at this point were

- Confirm common goal between stakeholders and designers.
- Have a plan, at least weekly, maybe even daily.
- Follow Tidwell *et al.* suggestion on how to understand how to design for people.
- Schedule third party reliant activities long before they have to be carried out.
- Make sure the designers have a design process.

6.3 Second Iteration

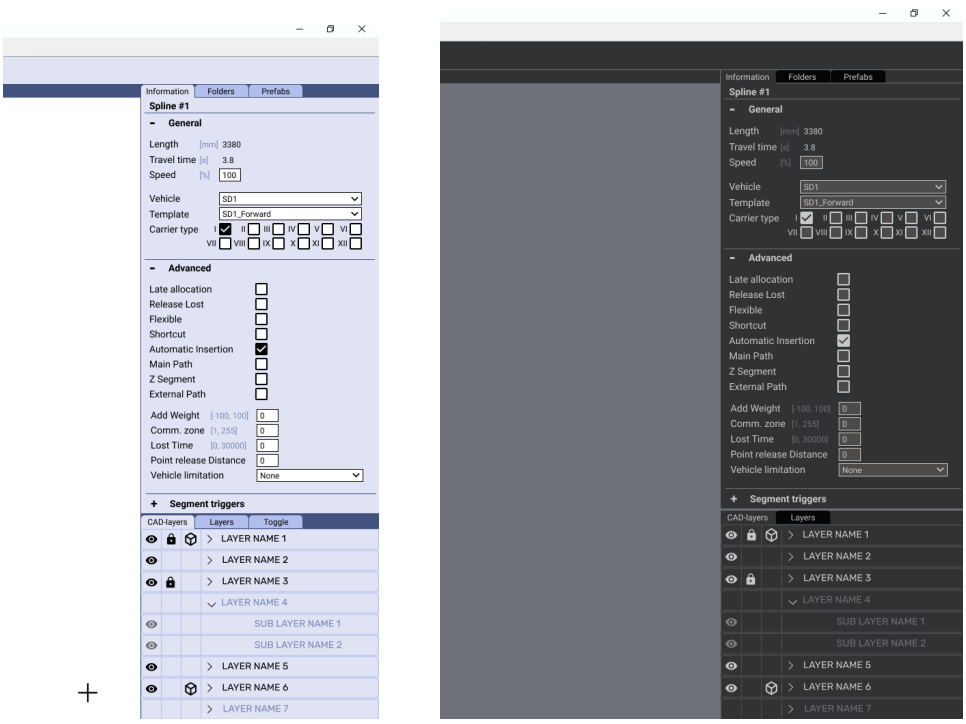
During the second iteration, we wanted to improve the main screen to give a more coherent version of our prototype in order to give our testing more basis for criticism, while also continuing to develop and evaluate individual elements to find out if they actually are viable. It is hard to give feedback on individual elements without the context, so getting more into the main screen was the main focus during this iteration.

6.3.1 Prototyping

The prototyping stage of the second iteration primarily depended on the feedback given in the previous iteration. We will dive deeper into elements, and give examples of the different versions that were evaluated.

6.3.1.1 Color exploration

It was during this stage of the prototyping that the emphasis on a dark mode had been requested. Several of the testers had previously mentioned that the light theme was an aspect that made it especially "old looking". Other than dark mode, a theme that matched with the color scheme of similar applications at Kollmorgen was briefly explored.



(a) **(b)**

Figure 6.22: Two of the color themes developed for Layout Designer.

The primary reason to have a bright and a dark mode, is to provide different kinds of contrast to the different user groups. As mentioned in an article by Justin Pot, dark mode is believed that it may reduce reading comprehension [35]. To accommodate for the different types of users, such as different eye sights or similar disabilities, the option to change from dark mode to bright mode should always be available, and the choice should be remembered.

6.3.1.2 Narrow Down Prototypes

The start screen in this iteration is similar to the previous one, with the most obvious changes being the color scheme and more exploration towards the starting of a new project since this was a previously annoying step for new users. Almost all new users discovered that the badly timed feedback in layout creation (described in Section 6.1.1.1) which was based in the ability to write your own "Drawing Name". The drawing name needs to be the directory to a CAD-drawing, which is the reason that we removed the option to write the path manually as can be seen in Figure 6.23.

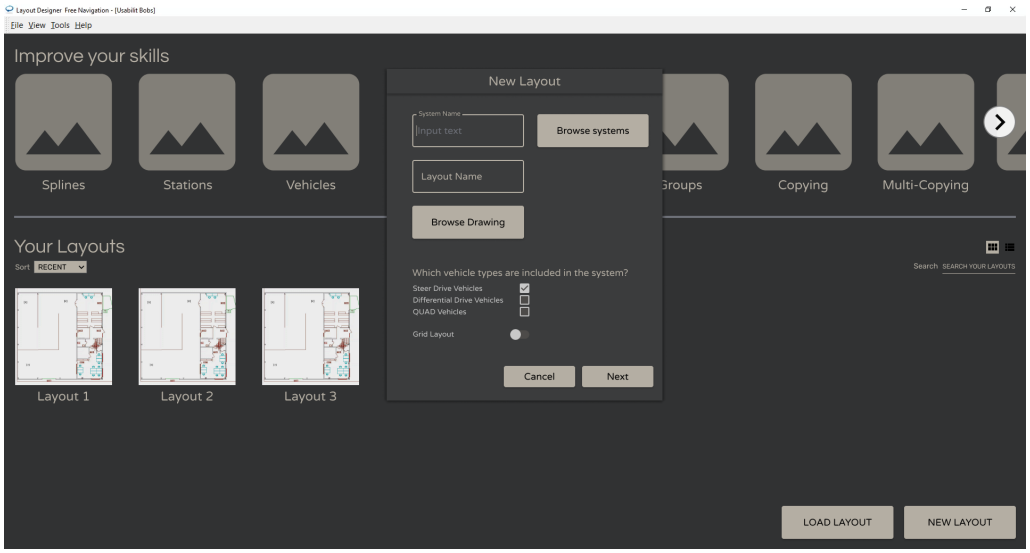
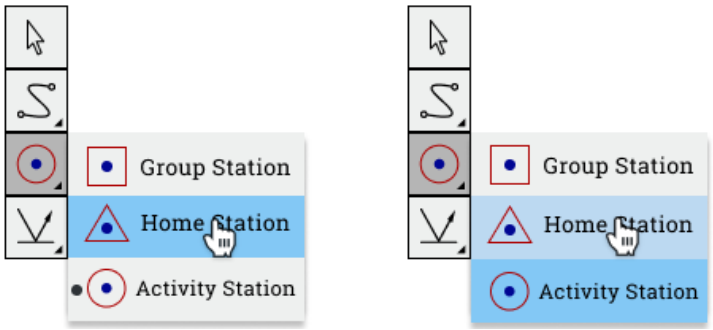


Figure 6.23: The first iteration of what the *New Layout* window should look like.

As can be seen in Figure 6.22a, more information is displayed on the right side of the screen to give a more detailed instruction of what information is expected to be present. In figure 6.22a the expected behaviour that is displayed is when a spline is selected. The current way of finding and editing this information is to right-click the spline you want to edit and selecting *properties*, which provides the user with a popup. The current version enables users to more efficiently change properties, while also allowing newer users to more easily spot what changes can be made.

Placing the layer access in a drop-down menu in the top-right (as can be seen in Figure 6.14) was generally disliked, and since the prototype was already invested in the tab functionality the most apparent and suggested alternative was to add it into a separate tab.

In regards to the toolbar, several new icons were created with the emphasis on having a coherent style convention. Several different types of selection and hovering alternatives were presented through questionnaires and interviews, and the alternatives in Figure 6.24 were the most commonly liked.



(a) **(b)**

Figure 6.24: Two of the most liked alternatives for hovering and selection.

Since Figma makes it very easy to crop and incorporate pictures, this was our go-to choice in the earlier stages. At first we thought Figma would mainly be used to combine elements from different programs but given our experience in the software compared to Photoshop and Illustrator it turned out to be quite efficient. Because of the efficiency we experienced within Figma, different layout alternatives and color palettes were tested in Figma as well.

Most of the prototypes will use a different toolbar, and most of the functionality will be hidden unless the tool is right-clicked. Our first example of how these tools could be sorted is given in Figure 6.25.

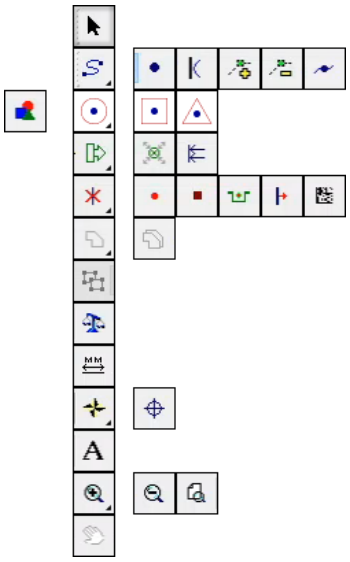


Figure 6.25: The middle column shows which tools are visible from the beginning, and the tools to the right of the middle column is the choices that will appear if right-clicking the tool.

6.3.2 Evaluation of designs

During this stage of usability testing, we were restricted to the comfort of our own homes because of the COVID-19. So even though we would have preferred to have these showcases and interviews on site, this is what we were limited to. We began with showcasing the suggested changes to our team at Kollmorgen, and continued with follow up SME interviews with two of our interviewees from iteration one.

6.3.2.1 Team Display

The usability testing of this stage was started with us showcasing the different screens that were currently made, as our team at Kollmorgen had not seen any of the progress this far. Many of the people in this team has had previous experience with Layout Designer, and has also been involved in previous redesign processes of Layout Designer. We did not want to view the previous attempts of redesigning as to not get influenced by design approaches that we had not taken part of.

Changes such as the dark mode was an appreciated change, as we asked the group if there was a need for dark mode several members exclaimed “Yes there is!”. Minor changes such as the placeholder pictures for the tutorials needed to be replaced, since it caused confusion while showcasing. Finally, it was explained that many of the solutions that we had found were shared with the previous redesign of Layout Designer, which was a good indicator that we were potentially on the right track.

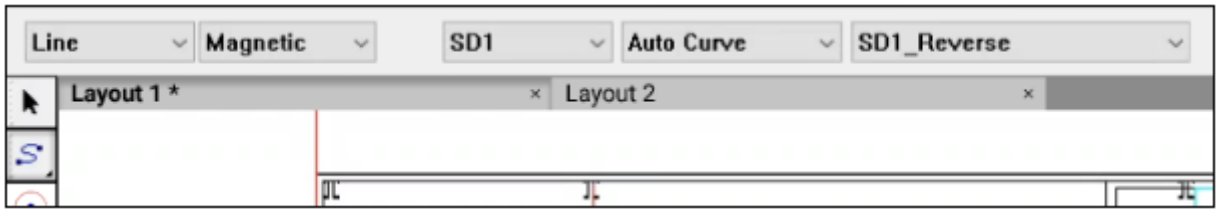


Figure 6.26: Spline tool selected with incorrect context sensitive functionality present.

6.3.2.2 Follow up SME interviews

The primary reason for us to reach out to the SME-users, was to decide if some of the suggested changes were welcome, unnecessary, or perhaps just not feasible to *actually* implement. One of the most important questions at this stage was in regards to the toolbar grouping and if it made sense, as much of the functionality was still not entirely known.

A remark that was given by one of the testers on the dark mode, was that the contrast was not good enough for his "old eyes". This confirmed one of our original thoughts that it was necessary to have *at least* two different color schemes, in order for everyone to use the software confidently.

Some of the most insightful feedback was regarding the icons, and in which way they were grouped (where our first suggestion was presented in Figure 6.25). A brief summarization from the two interviews we had concluded that our reflector tool needed to be adjusted, while most of the other tabs could remain *mostly* the same. Comments about how several different kinds of reflector's could not be simultaneously active, and how some of the navigational tools were directly linked to what system was available on site, gave us ideas of how to approach the next iteration more efficiently.

We had explored some alternatives to only show information that is connected to the currently selected tool on the top of the work space, as shown in Figure 6.18. The expert users liked this, stating that people will select a property and then forget to change the vehicle type or the type of segment function. One statement was that "This works like a prompt, to look at those settings to make sure they are right". It was also brought to our attention that we had misinterpreted the connection between some tools and their connected functionality. Shown in Figure 6.26 is the example we showed during the interviews, we were then informed that only two of the five alternatives were actually relevant for the tool. This was important to realize that we did not know as much as we thought about the context sensitive functionality and information we wanted to design for.

Some constructive feedback regarding different kinds of screen real estate were given. They stated that layers took up too much space. Configuring layers is only a one-time setup in Layout Designer, and giving it the constant screen space that software's

such as Adobe Photoshop has would be a waste. To them, the tutorials of the starting screen took up way too much space as well since they would practically never use them. The last consideration of the SME's were to be careful when occupying the horizontal view as the prefabs of Figure 6.14 shows, but the functionality of saving different configurations with the ability to transfer between coworkers were exciting.

A problem in visualising information that Layout Designer suffers from is when two or more paths are exactly on top of each other, the software will only display the one drawn last. Our assessment of this was that it was hard to know what had been done in certain areas of the layout as potentially a majority of the information could be hidden. Our approach to solving this was by showing all of the lines in different colors side by side, a solution that can clutter the layout quite fast. An idea we had never considered was brought up by a subject matter expert during our follow up interview, each line has an arrow on it to indicate its direction, these overlap as well, however, they could be moved and displayed in the correct color.

6.3.3 Guidelines

Even though these SME interviews took place pretty late into the project, all of the functionality related to the toolbar options was not clear enough for us, which made us waste time developing versions that were not viable. Our initial following of the guideline "Not showing anything to your stakeholder's that you are not happy with, they just might like it", made us second guess ourselves to an excessive level.

We have spent some time during this iteration just looking for the correct prototype to continue from and to showcase for others. This was unnecessary and largely an effect of our disorganized workspace in Figma. We could have saved a lot of time and made it easier to make a coherent design if we organized our prototypes better and created components to reuse when needed. A guideline regarding an organized design workspace does therefore feel justified. Not only the workspace should be organized but the plan as well. We mentioned this before, and we did have a weekly plan, but our plan did not include enough details. We sometimes felt like we were done, but when listing relevant parts we noticed that things were still missing. At this point we made a plan for each screen we were planning to make in the coming prototype phase and listed details to include in them respectively.

While designing for Layout Designer we often had to look back at at our findings and decisions. Which makes us believe that documentation of the communication is very important when there are so many elements to consider designing for. This documentation was relevant both for our internal communication and for our findings from showcasing our designs. With this we formulated the guideline "Continuous and documented communication is key".

During this iteration we started exploring colors, something that took time and that we found difficult. Especially the aspect of an interface with so many details in it

made the color exploration challenging. While just watching some color palettes independently they could seem intriguing but when applied to the interface it would not look good. We used multiple tools to explore color palettes, Colooors [36], Colormind [37], Palettte [38], Paletton [39], Coloring for Colorblindness [40], and Material.io's Color Tool [41]. We believe Paletton, Coloring for Colorblindness, and Material.io's Color Tool are the most useful when considering accessibility. For the color exploration to be more beneficial and efficient we suggest using a scrapped down version of your interface, do not include every imaginable detail at first. Use a version of your design that can showcase all essential parts of your design, with this it is possible to find a good base for how to organize colors and how to properly show the depth of the interface. Then the details can be added. So in short, have a template of your design that showcases only the essential parts, do not try to finish the whole interface in the first prototype.

To sum up the guidelines we had at this point were

- Confirm common goal between stakeholders and designers.
- Have a plan, at least weekly, maybe even daily.
- Follow Tidwell *et al.* suggestion on how to understand how to design for people.
- Schedule third party reliant activities long before they have to be carried out.
- Make sure the designers have a design process.
- Communication is key.
- Only display information relevant to the current choice.
- Be careful when occupying vertical screen space instead of horizontal screen space.
- Keep the design workspace organized.
- Have a detailed plan for designs.
- Have a template of your design that showcases only the essential parts.

6.4 Third Iteration

The third iteration consisted of refinement and clarification of the prototype, and then concluding the iteration by trying to determine what a relevant future iteration would contain.

6.4.1 Prototyping

Most of the changes in this prototyping iteration are focused on rework and refining instead of implementing or testing new concepts, as another iteration beyond this was very uncertain. But additions such as Figure 6.28 was still made, as it was deemed necessary by us to have it implemented eventually.

6.4.1.1 Toolbar

The feedback given in the previous iteration stated that some of the tools were placed incorrectly, so Figure 6.27 has attempted to correct that. There are now three very similar rows in this version, the reason behind this is because only one of them will be visible at a time. Only one reflector type is allowed to be present in one layout, but all of the other navigational tools are *allowed* depending on the warehouse. One of the rows from the previous iteration is completely removed and moved in with the same row as the reflectors, this is because they are all navigational tools and situational depending on what you choose in the setup phase. Some of the buttons that we thought were superfluous were re-added as they served very specific roles that made them important in certain scenarios. Buttons that could be triggered on and off unrelated to other selections were added to the bottom of the toolbar with a smaller size to differentiate them from the other toolbar buttons.

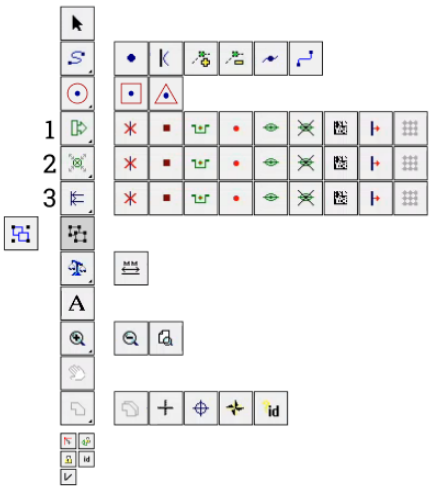


Figure 6.27: The toolbar groupings based on feedback from expert users.

Since some of the settings regarding the toolbar layout are determined at the start of the project, it may become necessary to adjust the toolbar in case a mistake was

made at the creation, so the user does not have to remake the project from scratch.

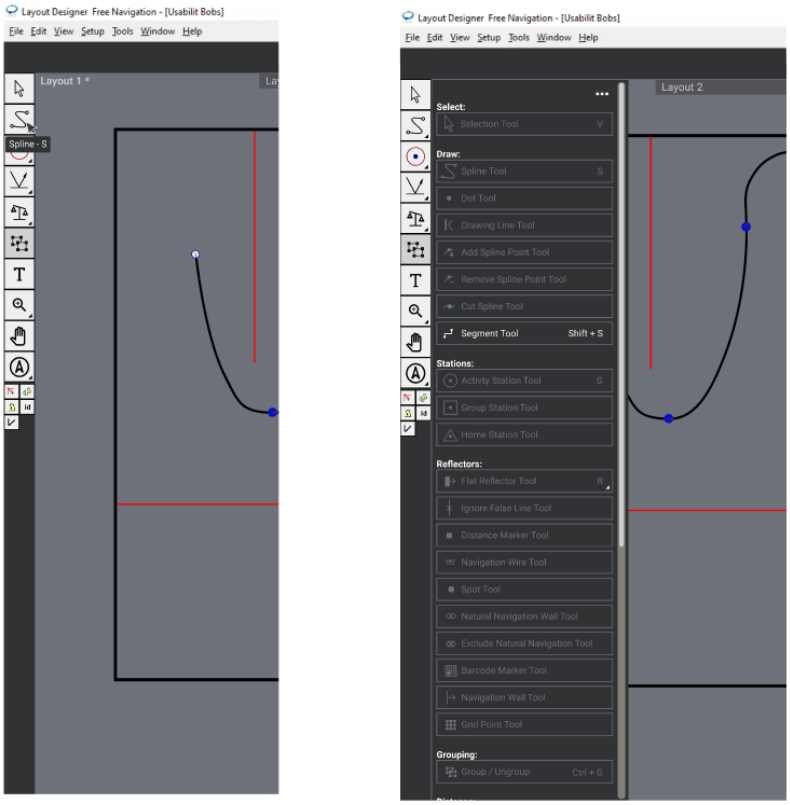


Figure 6.28: Shows how the toolbar expands, showing a detailed list in order to allow for advanced customization.

6.4.1.2 Additional improvements

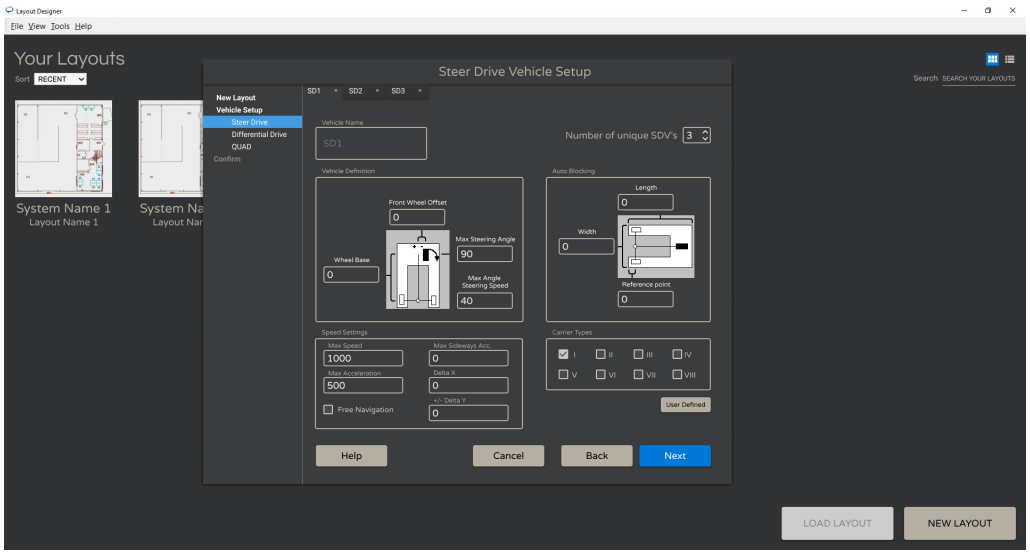


Figure 6.29: Improved the *New Layout* screen by adding a progress indicator.

Additional screen in regards to how to set up vehicles were added, with an added progress indicator to the side of the setup. Some of the earlier feedback of iteration one gave some hints towards an unclear setup stage. Adding a progress indicator as shown in Figure 6.29 allows the user more clearly sees which steps are completed and which ones remain were the bare minimum. These changes would need testing in future iterations, as they are not tested enough to be confidently added to the design.

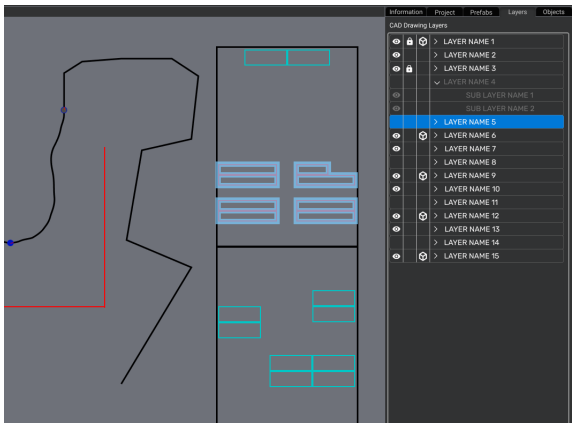


Figure 6.30: Clearly show how the background layer could be highlighted when selecting it in the layer-tab.

In the current version of Layout Designer, the only way of finding out which CAD-layers should be disabled, you need to turn them on and off systematically in the popup window. The SME's discussed that it was a one time setup, but that it could take several days to setup if the drawing was big enough. A feature like what is shown in Figure 6.30 would certainly quicken the process, but decisions such as highlight color have not been properly evaluated.

Having the possibility to drag the tabs is something that was considered a good idea by all of the SME testers since they have all mentioned them desiring the possibility to slightly change their layouts. In Figure 6.31 it is shown how the tabs would snap when being dragged around the work space.

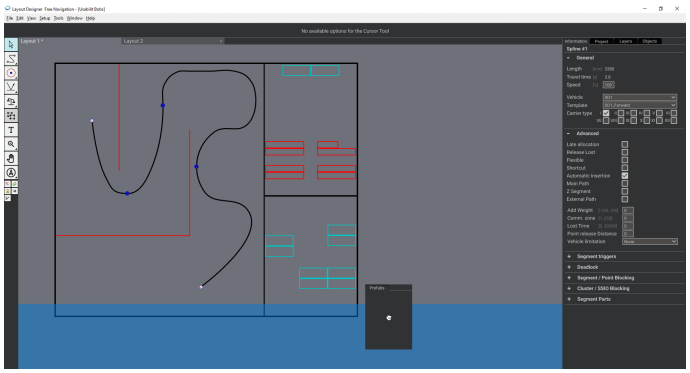


Figure 6.31: Shows how a dragged tab looks like, and in which way the tab is going to snap when released.

To accurately display what information should be visible when selecting certain tools is something that we have not evaluated properly, since we have struggled in previous iterations to connect tools and functionality, and will be discussed further in future work. In Figure 6.32 we display what should be shown when having the *text edit tool* selected, and in which ways highlighting could be further explored. For more examples of context sensitive information, we refer to Chapter 7.

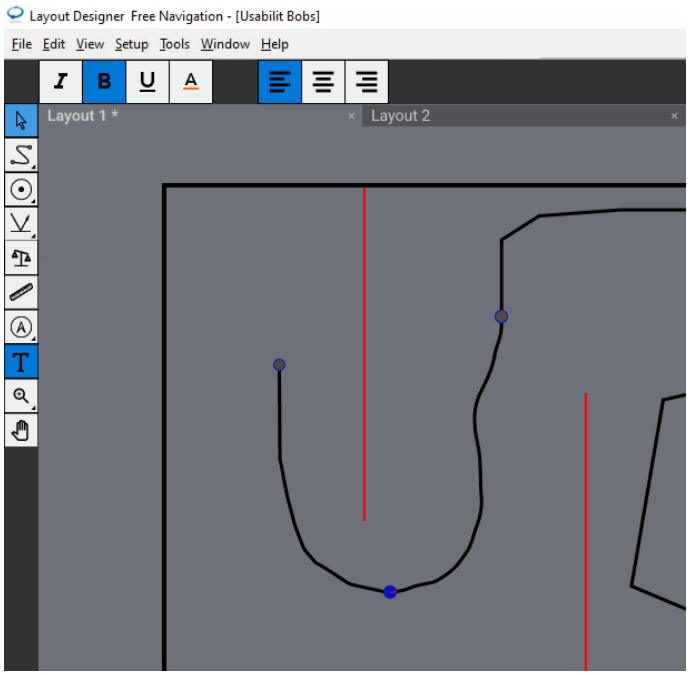


Figure 6.32: One example of how the context sensitive information may be shown.

Figure 6.33: Simplified colors for easier testing.

A different variant of the color scheme was made of the work space, to accommodate for less colors in total. In case the need for a different color scheme arises, the Figure 6.33 version will be easier to test.

Something that should have been developed during the early stages of our project was the style convention. We developed this when realizing that all of our components or interactions were slightly off, by either pixel or color. Figure 6.34 shows a brief overview of our visual guidelines. The style conventions are as shown mostly inspired from existing ones at *Material Design* [42], which is created by Google in an attempt to create an entire visual language. Just the evaluation of text field components in Material Design included hundreds of participants [43], which leads us to believe that they will likely be an improvement over something that we could design and research in this project's life span.

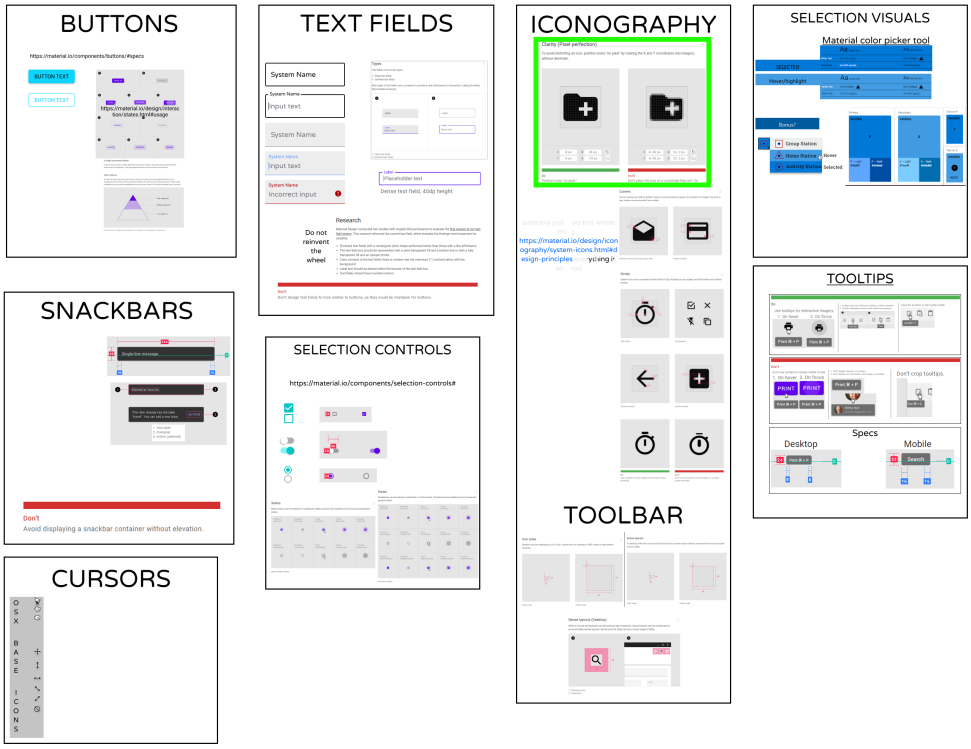


Figure 6.34: The collection of style conventions used at the end of the project (which can also be seen in Appendix C).

6.4.2 Evaluation of designs

Our initial goal for this final design evaluation was to make use of the prototyping tool in Figma and ask participants to perform similar tasks to those performed during the evaluation of Layout Designer (Section 6.1.3). This would have provided us with insights regarding if our design approach would make it easier to use the software for new users. Due to an extended prototyping iteration this was not done. Instead we showcased our designs for users of Layout Designer who work at Kollmorgen. We asked for their input about our different screens and illustrated ideas and we also requested them to share if they felt anything was missing.

The meeting was done via Microsoft Teams [44]. We shared a screen of our Figma prototypes and recorded both the screen and our conversation as we went through each of the screens. The reasoning behind recording both conversation and screen was to be able to more easily be able to understand which design and aspects of that design they were referring to when they gave feedback.

6.4.3 Guidelines

We did have plans for a style convention earlier than this iteration, but we did not make the correct assessment regarding its importance. We prioritized working on

the screens we thought were the most important. Because we had no universal style convention while designing these screens, we had some variation in how our components looked and had to modify already finished screens once the style conventions had been made. Because of this we suggest finishing the style convention early on to avoid doing the same work twice, as well as working with components that can be reused throughout the design.

When we started working with more details, we noticed that we did not know as much as we had assumed. We may have managed to provide an example of how the overall details should look independently, and some examples of when they are used together. However, what is lacking is a full fledged template on how settings and context sensitive functionality should be prioritized and grouped. This could have been avoided with more communication with the subject matter experts, we could have tried to gather more data about the most used tools and most used settings. So, when working with software with a vast amount of functionality, make sure you gather knowledge about what to prioritize and how to properly group the functionality.

While we personally liked the suggestions from Tidwell *et al.* on how to understand design for people we realise that this guideline is too specific and not very interesting. We chose to remove this guideline and instead suggest looking at other guidelines and design principles, in addition to our guidelines.

Our guideline regarding having a plan, and our guideline regarding having a plan for the design could be combined into one guideline, "Have a detailed plan of how to handle each screen".

We also realised that some guidelines could be made clearer by being rephrased. E.g. the guideline about only showing information relevant for the current choice has been rewritten to be more clear and also contain more information. It has been updated to urge the designer to properly group information and functionality throughout the design.

During our follow up interview with one of the subject matter experts we realised that the context sensitive grouping is valuable in more ways than to just conserve screen space. One of the subject matter expert said "I like that a lot. And the reason that I like that a lot is because people will select something and then forget to change the vehicle type or the type of segment function. This works like a "prompt", look at these to make sure they are right.". This gave more weight to our guideline regarding information and functionality grouping.

Our designs are sometimes of varying quality levels, often heavily correlated to our level of understanding for the functionality. We realised that it is important to have a proper understanding of the core functionality of the software to be able to know what to prioritize and what functionality to emphasize in different scenarios.

When testing our palette in an online tool, Coloring for Colorblindness [40], we realized how important it can be to convey information with other cues than only color. What we thought was a color clearly distinguishable from the base colors, instead appeared to be rather similar to the base colors for those suffering from Protanopia [45], as can clearly be seen in Figure 6.35.

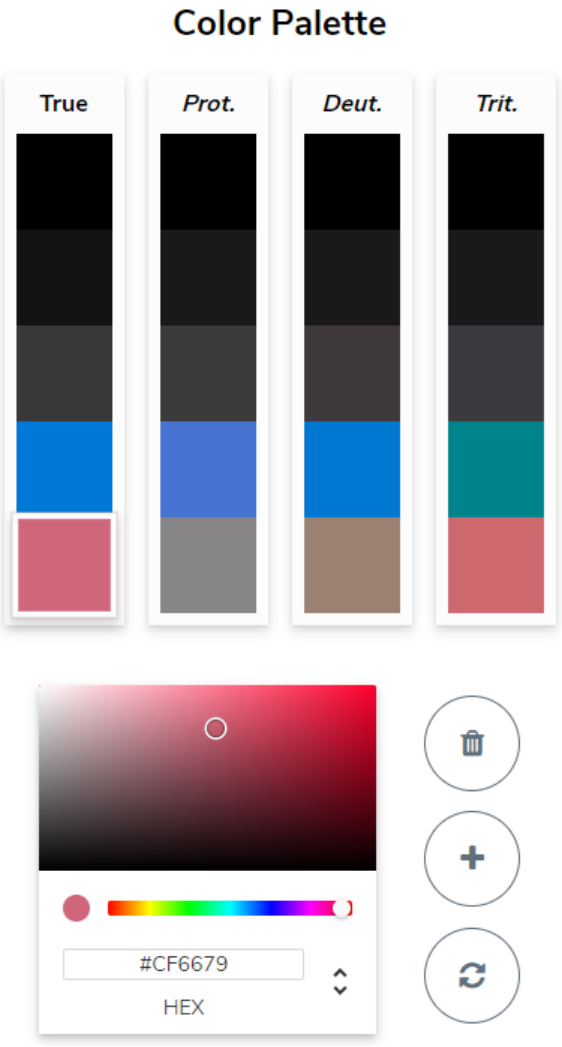


Figure 6.35: Illustration for how the same colors look different to people suffering from a type of color blindness.

The guideline "Try to convey important information with other cues than color" came from this realisation. An example of how we have implemented it can be seen in Figure 6.36.

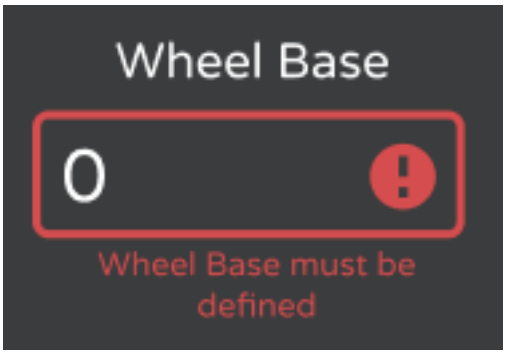


Figure 6.36: Example of how important information can be shown with text and symbols as a complement to the color.

At this point we realised that we did not have a sufficient amount of guidelines related to the design. We therefore took another look at the previous evaluations in an attempt to address potential usability flaws in Layout Designer and our design ideas.

When revisiting the feedback from novice users we remembered that they mentioned that a tutorial could be helpful, they also mentioned that they did not understand the functionality of some tools. Currently tutorials for Layout Designer does exist, but not within the software. Our idea to make the access to the tutorials easier was to include them in the start screen. An early design of this can be seen in Figure 6.16. However, we have also made a mistake. Layout Designer had a tool (shown in Figure 6.37 used to select an object, or a tool, to open the documentation related to that selection. This tool was overlooked by us as can be seen in Figure 7.9 where we have failed to include it anywhere in our toolbar. We highly recommend that this tool should be included as it is an easy way for the user to quickly reach the relevant part of documentation. With the help of Nielsen Norman Group's article regarding usability heuristics for user interface design [46] in which they mention "Help and documentation" we reached our guideline "Provide easy access to help and documentation when needed". Nielsen writes "Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large." but he also mentions that it is better if it is possible to use the system without documentation [46].

Figure 6.37: Tool used to access documentation about specific functionality.

One of the novice users stated that she "felt betrayed" after attempting to create a new layout and failed. A version of the provided error message can be seen in Figure 6.2. It states that a specific drawing could not be loaded. However, it does not clearly convey the information that the user can not provide a name for a new drawing and must instead provide the path to an existing file containing the drawing. The novice user did not now exactly how to remedy the error after receiving this error message. Another example of an unclear error message can be seen in Figure 6.3. In this figure it is presented as information but it does in fact remove the whole spline you are currently working on, which is undoubtedly in fact an error. Furthermore, it does this removal when the 39th splinepoint is placed so even after seeing this message the correct information has not been provided to the user. Bruce Tognazzini has written that "Error messages should actually help" [47], in combination with Nielsen's heuristic about matching between the system and the real world [46] we formulated our guideline "Error messages should be clear, with connections to real life".

We chose to remove the guideline "Make sure the designers have a design process" as we figured that it was a very limited knowledge contribution. Anyone designing should already know this and therefore it should not be needed to include it in our list of guidelines.

After our final guideline iteration, we ended up with the following guidelines. We chose to divide the guidelines into two categories. One category for the guidelines related to the process and one category for the guidelines related to the design. All guidelines are more thoroughly explained in section 7.3.

The **process** related guidelines are

- Continuous and documented communication is key.
- Confirm common goal between stakeholders and designers.
- Have a detailed plan of how to handle each screen.
- Schedule third party reliant activities long before they have to be carried out.
- Structure the design workspace and group related concepts.
- Have a template of your design that showcases only the essential parts.
- Create style conventions for your project early on.
- Confirm your understanding of core functionality before investing time in designing for it.

The **design** related guidelines are

- Provide visual feedback related to the software's status when relevant.
- Group information and functionality in a way that makes the user see the correlation between relevant elements
- Be mindful when occupying vertical screen space instead of horizontal screen space.
- Try to convey important information with other cues than color, especially when errors may have real world consequences.
- Provide easy access to help and documentation when needed.

- Error messages should be clear, with connections to real life.

7

Results

Our results consist of the prototypes that were developed during the course of the project, and the guidelines that should be considered when designing functionality-heavy software. Appendix B is of particular interest for employees at Kollmorgen, since that lists several of the current issues and inconsistencies currently present within Layout Designer.

7.1 Overviews of the prototype

There are two main screens of our prototype, one of the starting screen, Figure 7.1, and one of the main working screen, Figure 7.2. To give a better perspective of the rest of the pictures, the reader should look at these pictures before continuing to section 7.2.

Figure 7.1 displays the start screen of our prototype with the tutorial screen turned off (Figure 7.7 displays how it looks with tutorials active). Turning off the tutorials is a simple button click from the *View* menu bar, and is a choice that is remembered through restarts of the software. It is either possible to left-click the layout twice in order to open it, or left-click it once and follow that up with the, now highlighted, button *Load Layout*. There are two different ways to for layouts to get displayed to the user, either as a grid layout or as a list layout, in Figure 7.1 the grid layout is shown. It is possible to sort these layouts in different ways with the help of the dropdown menu, such as system name, layout name, recent and size, with the current selection being *recent*. The two most important attributes of a layout for a developer is the System Name and the Layout name, which is the reason that these two names are clearly shown no matter if the layouts are shown as a grid or a list. The option to search is available on the right side of the screen, which reacts instantly as the user types letters into the search bar. The search finds matches for both the *System Name* and the *Layout Name*. A miniature map of the layout should be shown of the Layout, allowing the user's eyes to more efficiently trace the screen. The button *New Layout* opens a large popup window that takes the user through the creation process of a new layout (displayed in Figure 7.7 (e)).

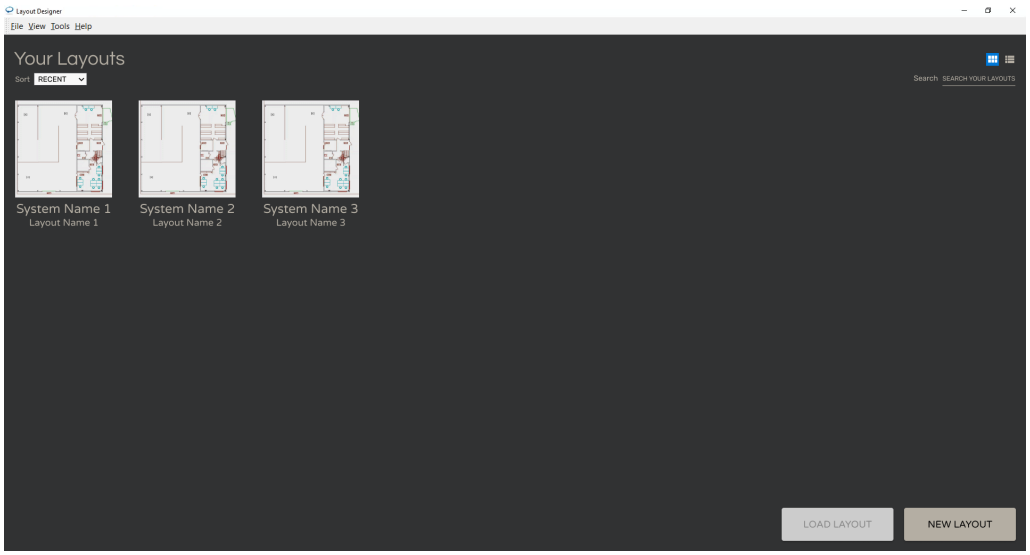


Figure 7.1: The start screen with the tutorials turned off.

Figure 7.2 displays a screen with two open layouts, with one visible and the other available through a tab. Currently there is a spline selected, which can be seen on the right of the screen as the informational tab showing information has spline information visible. Some of the information visible can not be changed through the tab, such as length or travel time, but other important attributes such as the vehicle or template is easily visible and accessible. The different headers of the information tab can be either minimized or expanded. The other tabs available to the right is *Project*, *Prefab*, *Layers*, and *Objects*. The project tab contains all of the different objects present in the Layout, such as the splines and the reflectors. The prefab tab would consist of user created "Pre-fabricated" objects, which could more easily be transferred between coworkers. The layers tab are related to the CAD-layer background which enables the user to efficiently change properties, a more in-depth look is given in Figure 7.5. The objects tab contains functionality that was requested by our subject matter experts, space has been allocated for the tab but since the functionality of it has not been properly evaluated it is not part of the final design.The currently chosen tool is the cursor tool, as can be seen by the blue marking in the toolbar and the text written in the *Context Sensitive Information* (that says "No available options for the Cursor Tool").

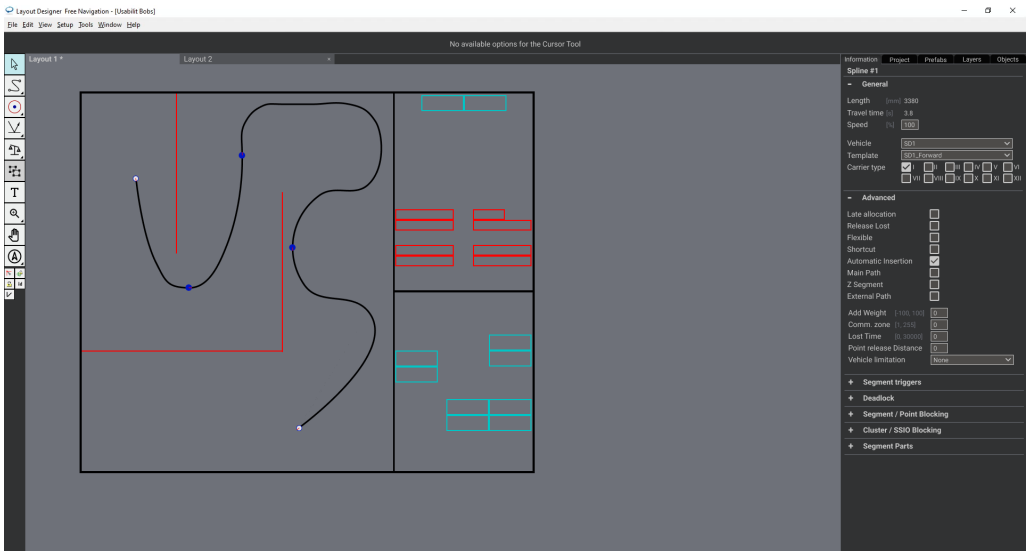


Figure 7.2: The work space visible with an open layout and a spline currently selected.

7.2 Summarized results and style conventions

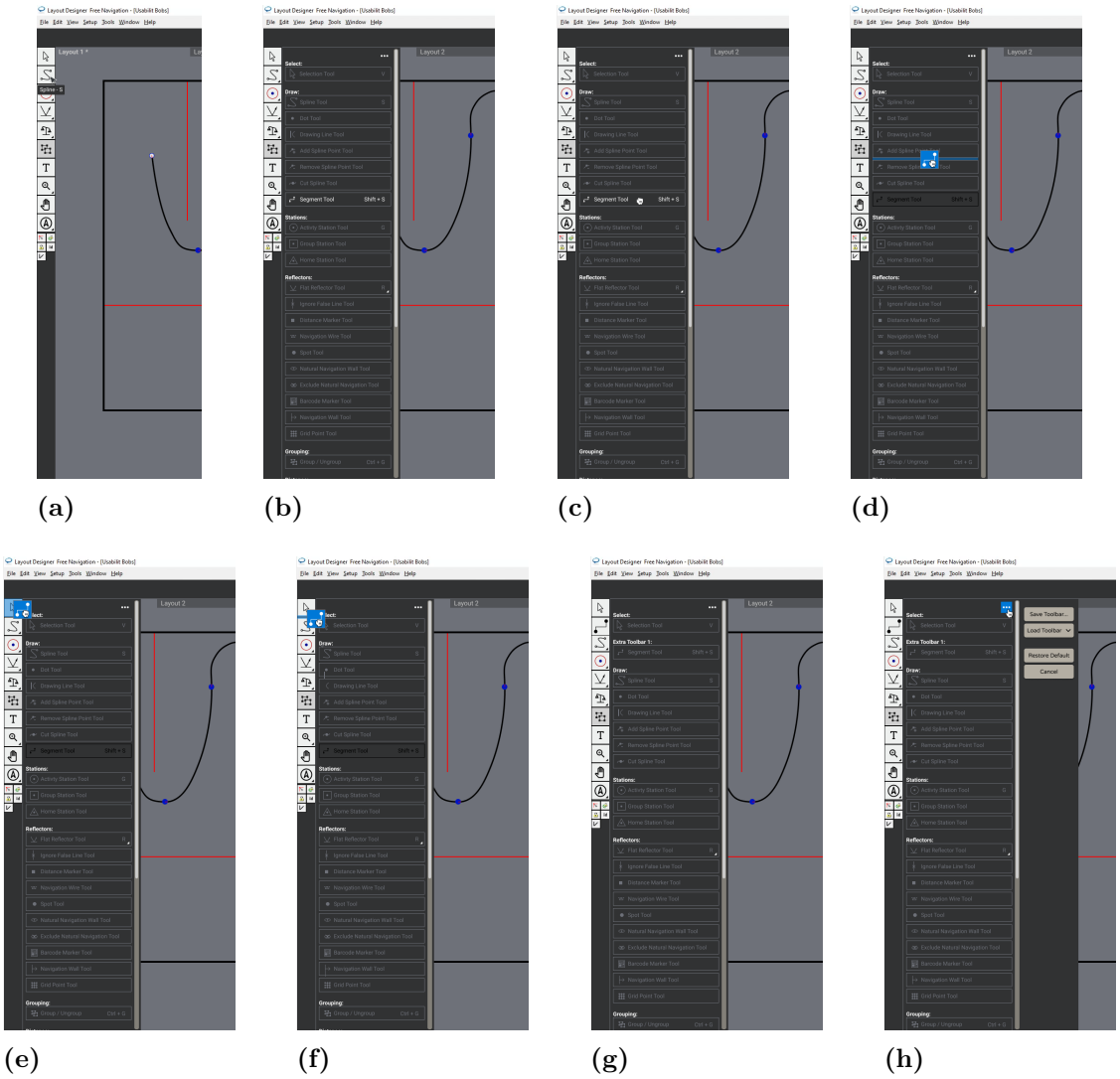


Figure 7.3: A way to redesign the toolbar after the user's specific needs.

Figure 7.3 (a) simply shows how tooltip's should be displayed for the toolbar, and what you can see in Figure 7.3 (b) to (g) is the process of dragging around segment tool into different locations. Figure 7.3 (d) and (e) simply displays how the user can make the segment tool into the "child tool" in a certain bar, while Figure 7.3 (f) and (g) displays that it is possible to create a completely new toolbar alternative. Different layouts of tools would be optimal during different stages of the layout creation, which is why 7.3 (h) shows that it is possible to save and load your current toolbar layout, and also have the option to easily reset it back to its' default configuration.

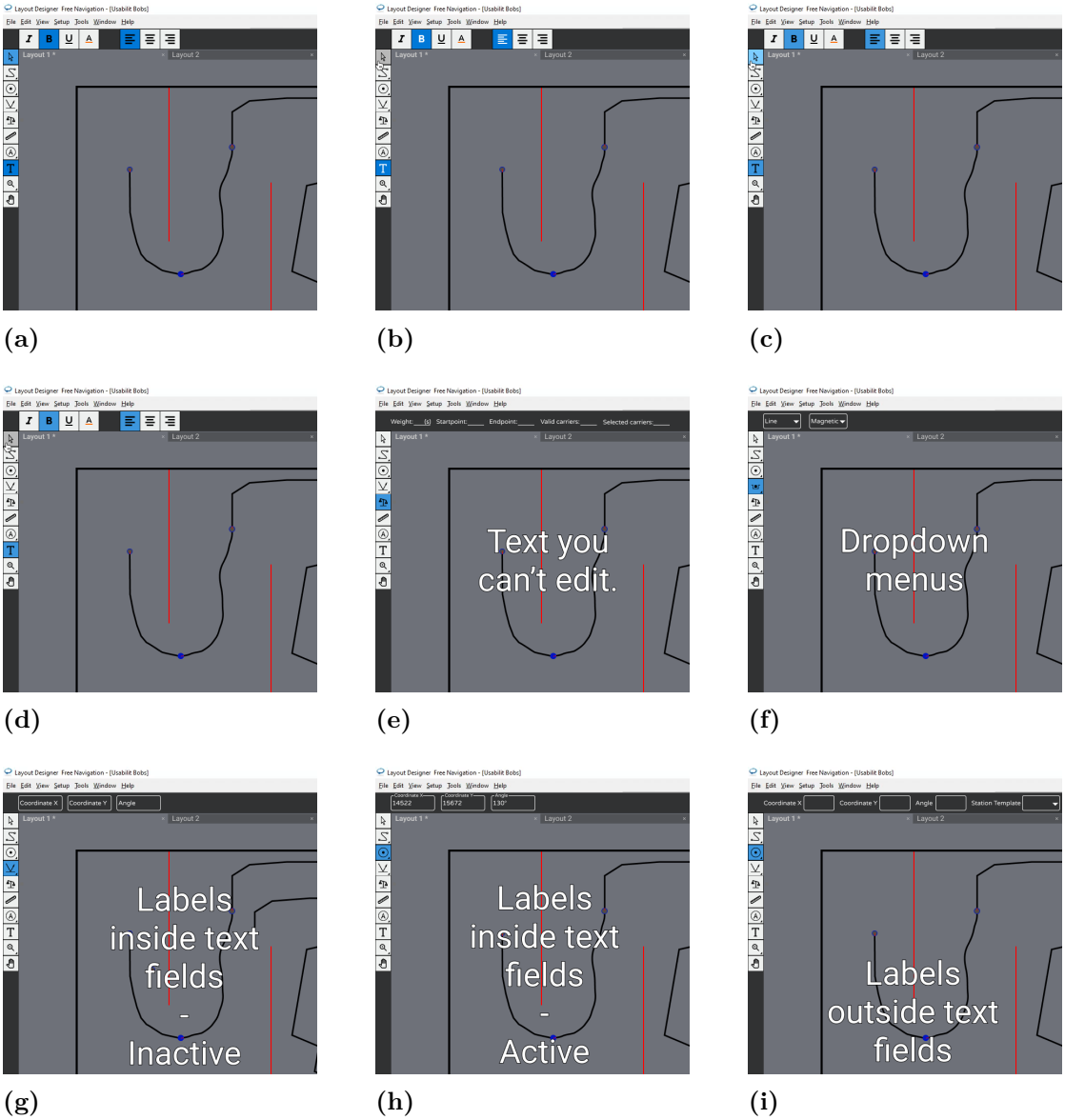


Figure 7.4: Examples of what information to show in the context sensitive upper bar, with added suggestions to explore further.

Figures 7.4 (a) to (d) shows different examples of how highlighting and selection could look like when having the *text tool* selected. Different options that is directly related to the text tool are visible on the top bar, this is what we refer to as *context sensitive bar*. Our limited research of selection and hover colors led us to believe that option (a) or (b) is the best suited for our current color scheme, but Figure 7.4 (c) and (d) serve as notable suggestions.

Figure 7.4 (e) shows how information can be presented when it can not be interacted with, (f) presents drop down menus. Figure 7.4 (g) to (i) present two different options for writable boxes.

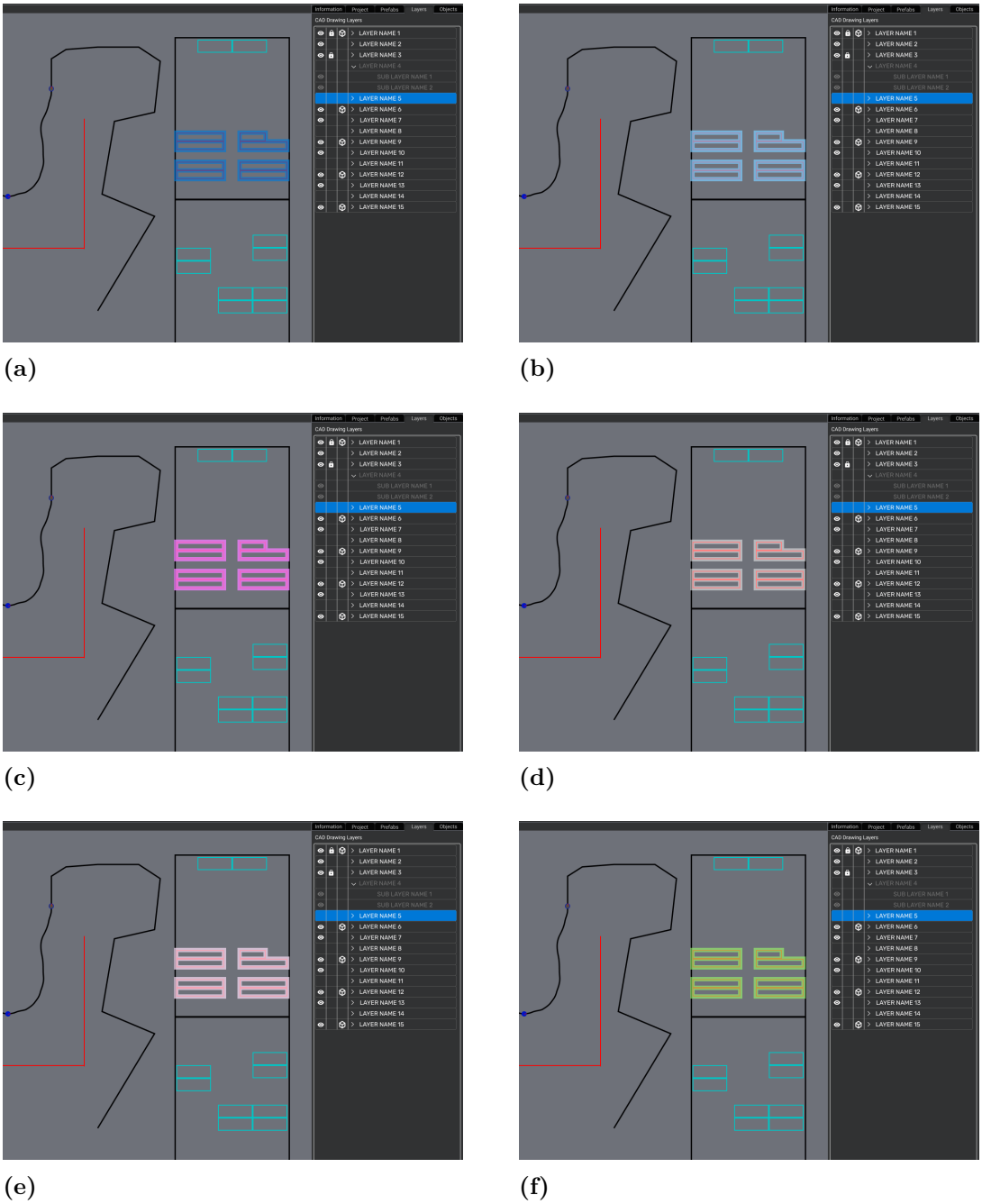
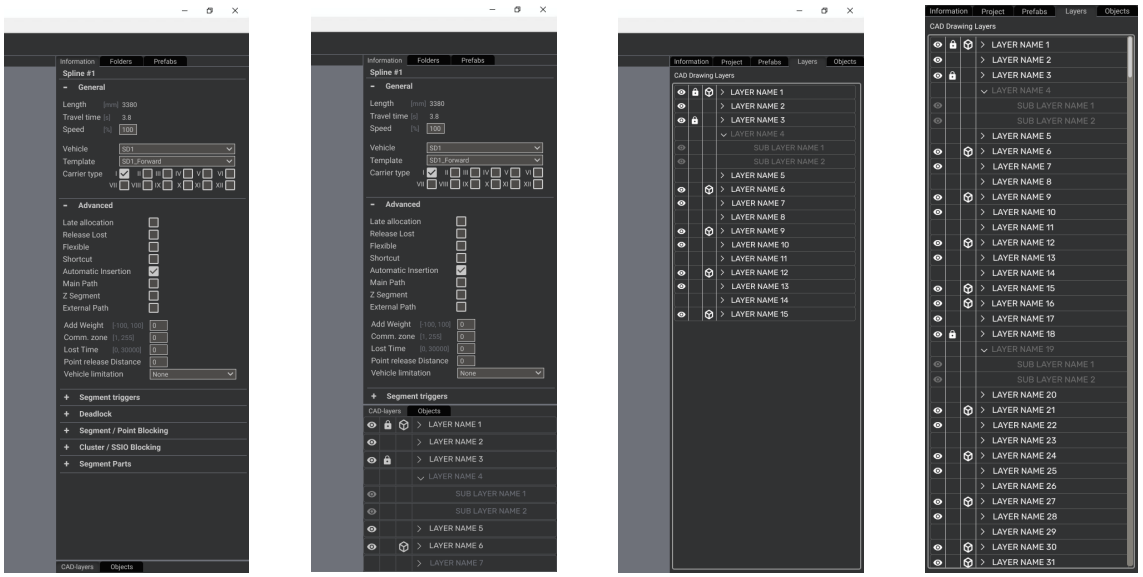


Figure 7.5: Examples of highlighted colors that displays which layer is currently selected.

Although not entirely researched in regards to which color is the most appropriate, Figure 7.5 attempts to explore, since the desire for this functionality to be present certainly exists. When selecting the layer in the CAD drawing the layer in question should light up or be bolded.

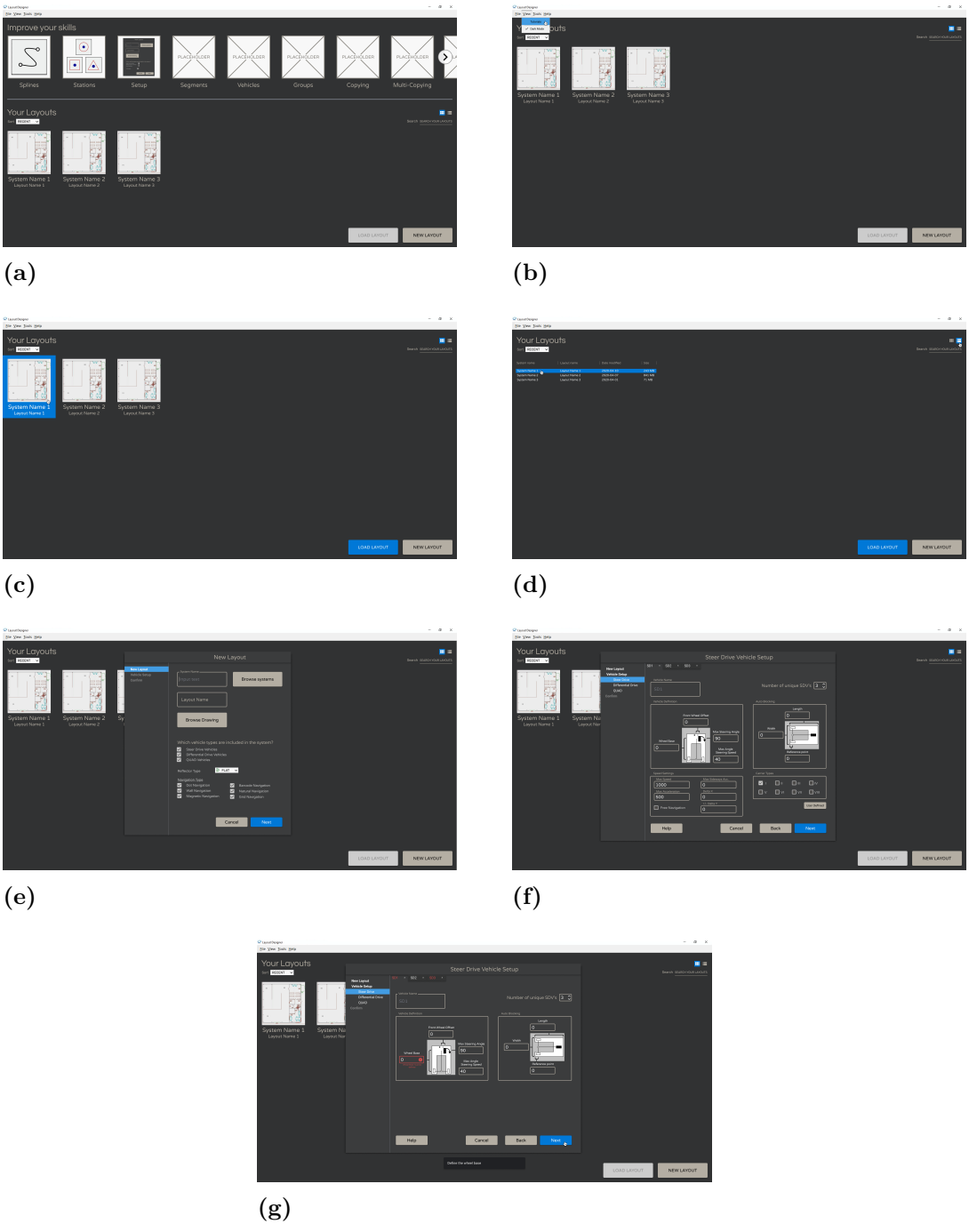


(a) **(b)** **(c)** **(d)**

Figure 7.6: Different positions of the *CAD-layers* tab.

Figure 7.6 (a) to (c) displays three different positions of the CAD-layer tab. During the course of the project *many* different opinions regarding which of these three is the best position has taken place. According to our discussions, Figure 7.6 (c) is the most practical one, as some users may not realize that the tabs are re-sizeable in (a) and (b). The different tabs should always be possible to close down by a right click action, since CAD-layer management is only necessary in the very first steps of the process.

Figure 7.6 (d) simply shows how a very long list of CAD-layers could look like, since it most likely would have a similar length in real life scenarios. This gives another reason to have the CAD-layers in its own tab on the side bar, to show a lot of information simultaneously.



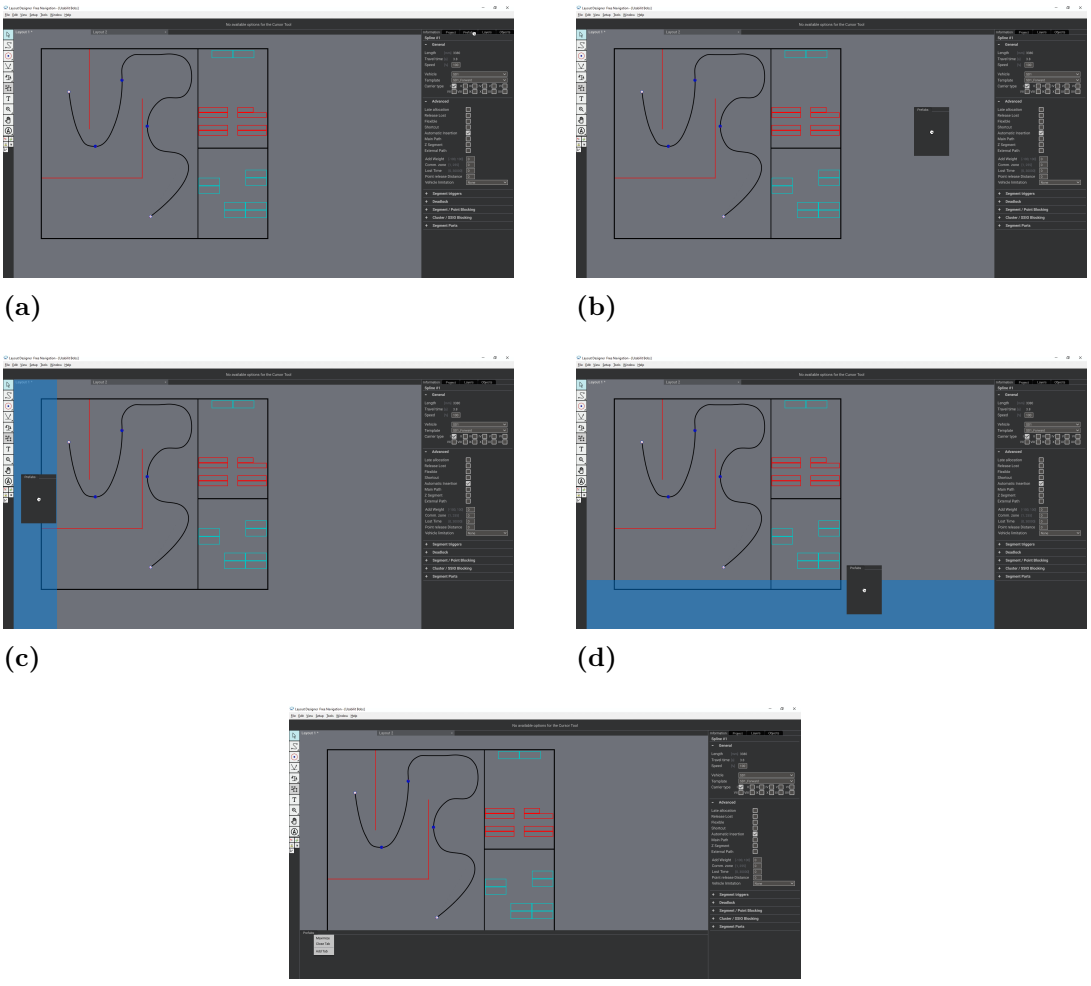
(a) (b) (c) (d) (e) (f) (g)

Figure 7.7: An overview of most scenarios of the start screen.

Figure 7.7 (a) shows the first screen that users meet, with tutorials on the top of the screen and your layout underneath. The option to search or to sort differently exist, in case the user wants to. Figure 7.7 (b) shows how to turn off the tutorials, and when the user makes this choice it should be remembered. Figure 7.7 (c) and (d) both show selection of a layout, with (c) showing layouts in a grid and (d) showing the same layouts in a list. The option to double-click to open a layout is possible,

but the *Load Layout* button also lights up when selecting one of your layouts. Figure 7.7 (e) and (f) shows the first two screens when creating a new layout. Some notable changes from the current Layout Designer setup stage is that the reflector type and navigation types are selected in this stage, and regulates which kinds of reflectors or navigators are available to the user when designing. Breadcrumbs have been added in order for the user to know where in the process the currently are. The ways in which error feedback is given has also been improved since it is now instantly shown, instead of the previous (very disliked) way of allowing the user to continue to the last screen and get an error popup there.

Figure 7.8 (a) to (d) shows how dragging the *Prefab Tab* around the working screen looks like. As the tab gets closer to a valid dropping point, the area of where the bar will snap to will be displayed in a blue tint. Figure 7.8 (e) displays the left mouse button being released and also displays what options are visible once right clicking on a tab.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 7.8: A display of how tab dragging should look like.

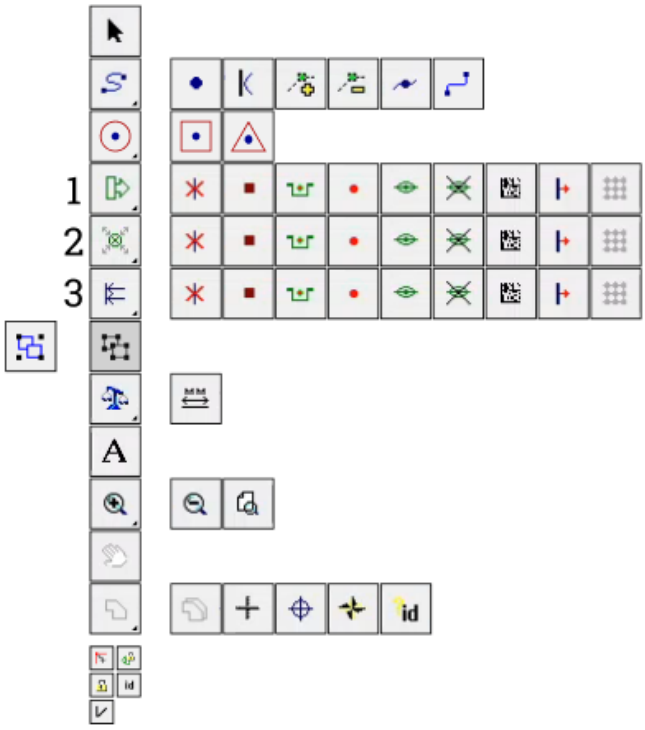


Figure 7.9: Our suggested toolbar grouping.

With regards to the feedback that we got from our SME interviews, this is the grouping of tools that our final iteration ended with. The grouping in Figure 7.9 is visualized with the old symbols. The tools to the right of the main symbols is what unfolds as the user right-clicks the tool. The grouping tool has the other symbol to the left, and simply shifts symbol corresponding to if the user has selected a group, or if several un-grouped items are selected. The smaller symbols at the bottom are switch buttons, which is the reasoning of why they are visually different from the rest of the buttons.

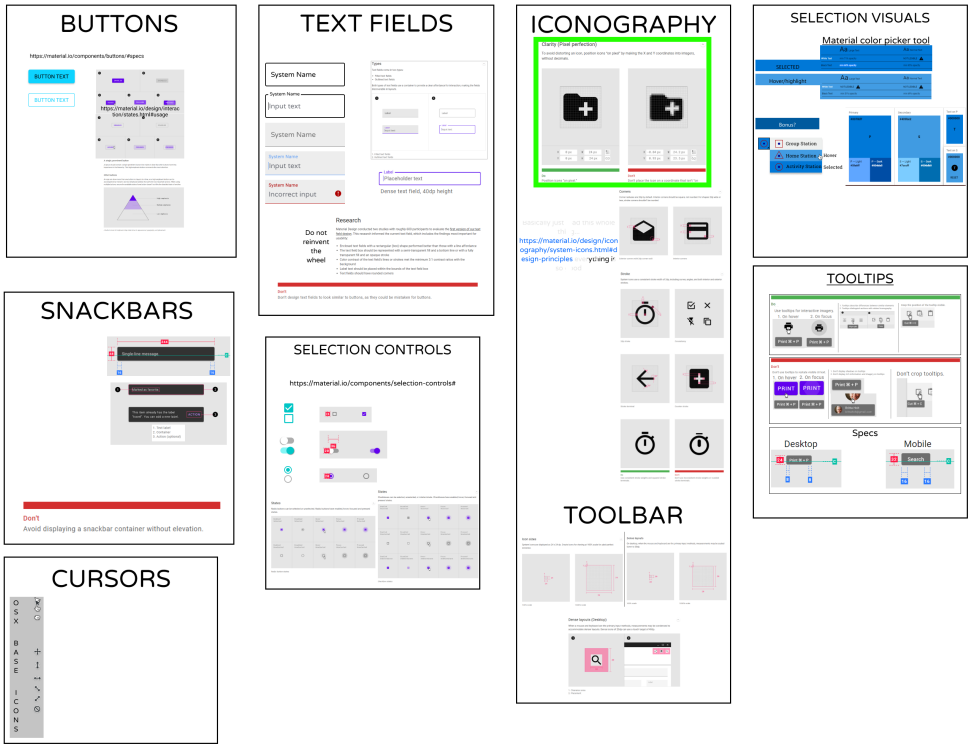


Figure 7.10: The collection of style conventions used at the end of the project (which can also be seen in Appendix C).

Figure 7.10 shows the nine different categories of style conventions that we created to aid the designing the new Layout Designer. The different categories include *Buttons*, *Snackbars*, *Cursors*, *Text Fields*, *Selection Controls*, *Iconography*, *Selection Visuals*, and *Tooltips*.

The least specific of the style conventions is the selection visuals since there are several different colors that the layout consists of, and the selection color must correspond in accordance to that specific background color. Iconography is mostly related to having a consistent thickness and spacing of the icon, in order to maximize readability. The text fields consist of a couple different alternatives, where we went with the two upper ones in most cases. Slight alternatives can (and have) been made in order to adjust to the current screen space. The different cursors are shown in different scenarios, in order to give the prototype realistic system response. The other style conventions in Figure 7.10 are mostly reasonable advice provided by Material Design [42]. The style conventions are followed in order to make a coherent prototype.

7.3 Guidelines

As the project developed, so did our guidelines. The guidelines that are visible here were being added successively after each iteration. If the reader wants to get a deeper insight to why these guidelines were chosen, we refer to the *Guidelines* sections of each iteration in Chapter 6. A description of what each of the guidelines mean in greater depth will be given in the subsections below these guidelines.

The guidelines identified are not very specific to the design of functionality-heavy path drawing software. The guidelines we have identified will likely work for designing functionality-heavy path drawing software and other software as well. We have however stated that some of the guidelines do become more important as the complexity of the software increases. The main reason the guidelines are not very specific towards designing functionality-heavy path drawing software might be because the domain is not unique enough to need very distinct guidelines. We do however believe that the complexity of a software changes the importance of certain guidelines and the domain can certainly affect which guidelines that should be focused on. During this project, and while formulating these guidelines, more well known guidelines and design principles were read. For us Nielsen's 10 Usability Heuristics for User Interface Design[46], Cooper *et al.*'s About Face[7], Tidwell *et al.*'s Designing Interfaces : Patterns for Effective Interaction Design [3] worked well when we were designing for functionality-heavy software. We suggest looking at other guidelines, in addition to our guidelines, as well.

The final guidelines are divided into guidelines regarding the **process** and guidelines regarding the **design** for clarity. The guidelines are not presented in any specific order.

The **process** related guidelines are

- Continuous and documented communication is key.
- Confirm common goal between stakeholders and designers.
- Have a detailed plan of how to handle each screen.
- Schedule third party reliant activities long before they have to be carried out.
- Structure the design workspace and group related concepts.
- Have a template of your design that showcases only the essential parts.
- Create style conventions for your project early on.
- Confirm your understanding of core functionality before investing time in designing for it.

The **design** related guidelines are

- Provide visual feedback related to the software's status when relevant.
- Group information and functionality in a way that makes the user see the correlation between relevant elements
- Be mindful when occupying vertical screen space instead of horizontal screen space.
- Try to convey important information with other cues than color, especially

when errors may have real world consequences.
- Provide easy access to help and documentation when needed.
- Error messages should be clear, with connections to real life.

7.3.1 Continuous and documented communication is key

As you explore the designs you get new insights, find new solutions that work better than previous approaches etc. It is important to share these with your team, make sure that everyone is in agreement when pivoting towards a new design approach.

Practically all communication between us as designers and between third-party members were documented. This allowed us to individually look back on made decisions, and made it possible to maintain our continuous communications and pace without it becoming bothering for the other designer.

7.3.2 Confirm common goal between stakeholders and designers

Our purpose of the re-design was clearly stated during the beginning of the project, to make it easier for new users to understand Layout Designer and at the same avoid affecting the efficiency of expert users in a negative way. To have this rule to fall back on made the process easier for us designers.

To follow this guideline, do not just assume that there is an understanding, you might have different interpretations of certain vocabulary. Ask questions and explain your own understanding of the goal so that it can be confirmed.

Without everyone having a clear understanding of the goals there are potential risks such as the designers spending time on an idea that does not at all align with the wishes of the stakeholders.

7.3.3 Have a detailed plan of how to handle each screen

The plan should consider both the process and the design. It is important to know when to do things, such as when to prototype and when to evaluate the designs but it is just as important to know more specifically what to do during these phases. Having a list of everything that should be designed in each iteration respectively makes it easier to divide work between team members but also to know if you will be able to deliver things on time.

During the project there was a temporary decrease in productivity, this occurred when there was a prototyping phase but exactly what to prototype during the phase had not been planned. Once a plan for what to prototype during that phase had been established the productivity increased again. More specifically, the plan was established through *Trello* [48], where a detailed list of each needed screen was made.

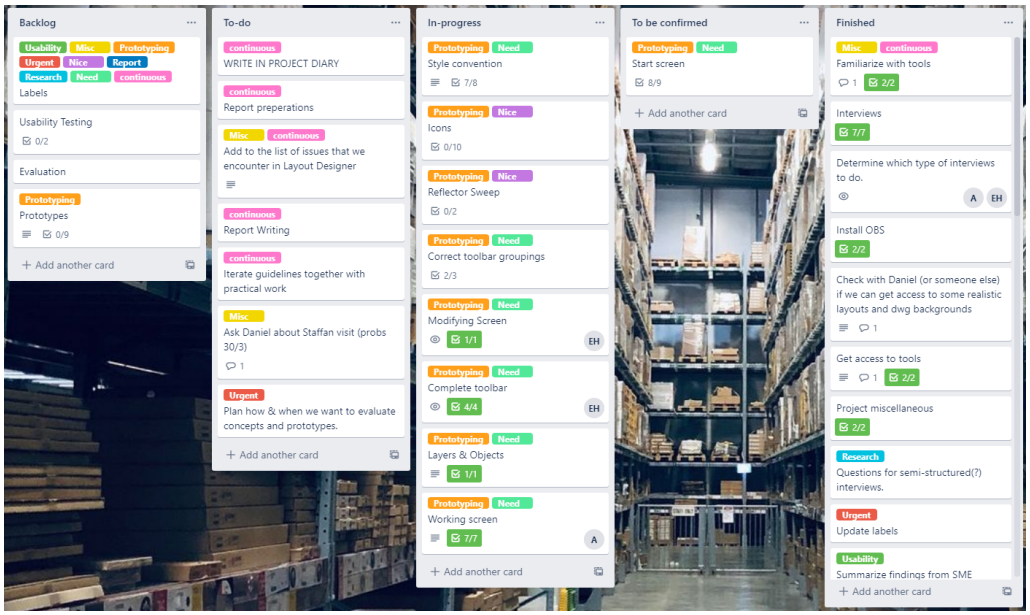


Figure 7.11: A more detailed list regarding specific screens that needed to be developed, and some that would be *nice* to have that had a lower priority.

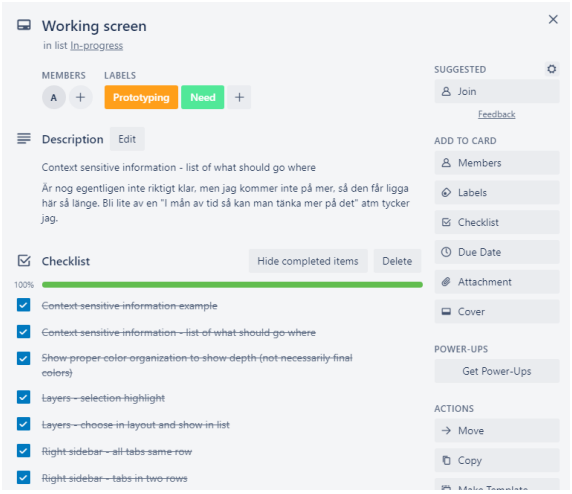


Figure 7.12: The progress of our working screen prototype, showing which details needed to be implemented before it was considered done.

When following this guideline, do not be afraid to modify the plan, the most important part is not always that everything goes strictly according to plan, but that everything that has to be done is tracked. A big benefit of following this guideline is that it helps with the guideline regarding scheduling third party reliant activities in time.

7.3.4 Schedule third party reliant activities long before they have to be carried out

This guideline is reliant on having a detailed plan. As soon as the designers know of appropriate times to receive feedback from third parties, they need to make an effort to plan this with the desired participants as soon as possible.

The importance of this changes depending on how scarce the target audience is. If the audience is very wide, it might become easier to recruit participants for tests with shorter notice, as there are potentially more people to choose from. In this project the observations with users in their natural work environment took weeks to book, and eventually these observations were cancelled due to COVID-19, then it took another couple of weeks to get the needed contact details for a questionnaire instead. It had not been considered that it could take so long, if this guideline was followed, an attempt to book the observations could have happened much earlier and would most likely have resulted in the observations actually being executed.

7.3.5 Structure the design workspace and group related concepts

To keep the design process efficient it is important to have a structured design workspace. It is easy for other team members to miss a certain design detail if it is not clearly shown in its relevant context.

The importance of this guideline increases depending on the complexity of the software to design and the number of designers who are cooperating. If there are many different screens and aspects to design for, it can become a time consuming task to navigate to the correct screen if there are no overarching groups or themes the designs can be placed into. Early on in this project no clear groups were created, that lead to more time consuming preparations for evaluations of the designs and it made it harder to keep track of what had been done.

When following this guideline, especially if more than one person is working on the project, do not regroup too often. When a member has gotten used to finding a certain design in a specific place and group, and that group is gone or has been moved, it can once again become hard to find what one is looking for. With that said, do not be afraid to reorganize when needed, a group that constantly keeps expanding can once again become an area that is hard to navigate.

7.3.6 Have a template of your design that showcases only the essential parts

With software of this complexity it is important to design from the bottom up. There are potentially too many details to consider at the same time initially. Start by creating a base layout that you feel comfortable with and then add details in order of importance. Some things might be reasonable to hide if they are not used

enough to justify them having a permanently visible position in the user interface. If needed, modifications can be made to base layout.

This guideline effectively describes how we created our base layout, starting with the workspace and expanding by adding things successively, such as the toolbar and the informational tab. Working in this way makes it relatively easy to undo a design decision. Some design decisions make sense in the current context, but then lose value as more things are added. An example of how we lost value over time would be our color prototyping. Since the screens that we tested our colors on were very detailed with around 20 (very similar) colors, each test took a lot more time than it should. After we ultimately reduced the amount of colors to 5, the testing time got drastically reduced. This screen is shown in Figure 6.33.

7.3.7 Create style conventions for your project early on

A style convention for components that are widely used throughout the software is important. It allows for re-usability and more efficient designing of following screens. Making the style convention early on makes it possible to avoid having to redo details in multiple designs at a later stage. Eventually, we created a template of relevant style conventions, with connections to color and overall functionality. These guidelines can be seen in Figure 6.34.

During this project it was a reoccurring topic how something should look, often changing between different designs. Once the style conventions were in place those could be looked at and the designing became much more streamlined. It is important that the style conventions for the project have some kind of confirmed validity. In this project they were almost entirely based on guidelines for design by Material Design [42]. This also falls in line with the suggestion by Cooper *et al.* "you should obey standards unless you have a good reason not to" [7, p. 124].

7.3.8 Confirm your understanding of core functionality before investing time in designing for it

To be able to correctly emphasize information and functionality it is important to understand the core functionality you are designing for. The importance increases as the complexity of the software increases since it poses the risk of having to redo more work if emphasis has been given incorrectly and functionality has been grouped incorrectly.

This guideline differs depending on if it is a redesign of an existing software or if it is a design of a completely new software. If it is an already existing software there are most likely expert users of it, they are a valuable resource regarding which functionality is essential. If it is a completely new software the "Confirm common goal between stakeholders and designers" guideline becomes more important, as a very clear set of requirements for the software needs to be defined.

As mentioned in 7.3.10 an incorrect functionality grouping was presented (shown in Figure 6.26) to the subject matter experts, this was because of lack of knowledge about the functionality. If this guideline had been followed better during this project, time could have been saved, and better feedback could potentially have been gotten if correct examples were shown.

7.3.9 Provide visual feedback related to the software's status when relevant

Generally when a software, for example, is loading, accepts input or an incorrect action has taken place, it is important to show that visually so the user understands that *something* is happening. The main cases where this can be seen in our design is in Figure 7.5, 7.8 and 7.7 (c). Figure 7.5 clearly shows how visual feedback related to which layer has been selected, and 7.8 indicates where the tab is going to end up when released. Figure 7.7 (c) simply refers to the button being highlighted when a layout is pressed once.

In Figure 7.7 (g) you can see a potential flaw within our design. This is because the setup indicates that the *next-button* is clickable, but only triggers errors when pressed. This displays very clearly what button needs to be pressed in order to proceed, but almost tricks the user into activating the errors. Having the error visible at all times would also be counter intuitive, since the user has not done anything wrong yet, so it is ultimately a fine line that needs to be explored. Since the blue button in Figure 7.7 (c) means continue, and the blue buttons in Figure 7.7 (e) and (f) does not *quite* have that implementation, we have not followed the guideline by Nielsen regarding consistency and standards which states "Users should not have to wonder whether different words, situations, or actions mean the same thing" [46]. A slight reevaluation would be needed for these screens in further redesign. A potential fix could be to give the button the highlighted color, and in that way its emphasis, after the required fields have been filled. The required input fields and the non required input fields could also be distinguished from each other. These two solutions have not been evaluated by us but could be worth to consider.

7.3.10 Group information and functionality in a way that makes the user see the correlation between relevant elements

It is always important to group information and functionality in a way that is easy for the user to comprehend. It becomes more important in functionality-heavy software because of the vast amount of potential information it can be confused with. Only give emphasis to available options and do not overwhelm the user with information that is unrelated to the current operation.

In Figures 7.4, and 6.21, there are examples of information and functionality grouped according to the current objective of the user.

To be able to correctly follow this guideline it is also important to make use of the "Understand core functionality before investing time in designing for it"-guideline. If there is a lack of understanding regarding which information and functionality that is actually related to the user's current task some things might end up missing or be prioritized incorrectly. In Figure 6.26 functionality which is not related to the current tool has been added due to lack of knowledge.

7.3.11 Be mindful when occupying vertical screen space instead of horizontal screen space

It is important to give sufficient space to the work area. The vertical screen real estate is more scarce than the horizontal and it is therefore important to be mindful when making use of it to not occupy more space than can be justified.

During the first iteration of subject matter expert interviews it was mentioned that the current toolbar in Layout Designer was often modified by the user, one reason, mentioned by a subject matter expert, was to increase the amount of screen real estate dedicated to the work area. This is a valid reason to be mindful with how the screen space is used by default to not force the users to always modify the layout to achieve adequate work space. In Figure 6.33 the layout shown for this project is visualised. This approach provides users with a big work area while still supplying the user with much information by making use of the horizontal screen space.

When following this guideline it is important to keep in mind that information will often be presented in tall and narrow windows, that needs to be considered while designing. If the guideline is not followed there is a risk of giving the user a work space that is too small, at least in the vertical direction.

It is also important to follow Tognazzini's principle regarding autonomy, stating that designers should "Enable users to make their own decisions, even ones aesthetically poor or behaviorally less efficient" [47]. Making the tabs adjustable as shown in Figure 7.8 provides good accommodation options for the user.

7.3.12 Try to convey important information with other cues than color, especially when errors may have real world consequences

As can be seen in Figure 7.7 (g), we have attempted to visualize errors in more ways than just changing the colors to red. Although red is probably enough of an indication for most people, color blindness should be considered. The different techniques used include the color red, symbolism (with the use of an exclamation mark), and a descriptive text (under the error and also in the snackbar).

It could possibly become problematic if all of the different techniques above were used each time an error occurred in Layout Designer, as it would become cluttered. Errors can occur in situational occasions such as creating splines, that is in the

current version of Layout Designer *only* visualized by changing the color of the nonfunctional part of the spline. During the last evaluation this visualisation was questioned by a participant. She had personally experienced issues with this, which is a good indication that an additional cue is needed.

The number of different splines that follow the same path could often be four or more, and all of the arrows and lines are placed in the exact same location. To visualize this more clearly, it has previously been discussed in Section 6.3.3 that arrows could be spread out slightly to give a decent overview of the overlapping lines, or the lines could be placed next to each other. Figure 7.13 shows how lines could be places next to each other. The approach of lines placed in parallel was generally liked during our evaluations. The suggestion about spreading out the arrows came from one of the subject matter experts and has not been evaluated. The solution could however be better than parallel lines, to reduce clutter, but it has to be evaluated before the quality of the idea can be determined.

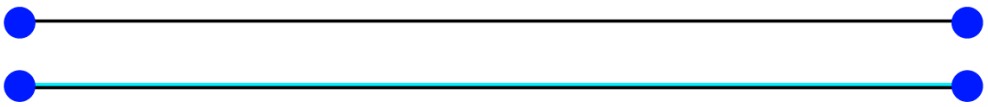


Figure 7.13: The first line shows the current visualization in Layout Designer. The second line shows our suggestion of how two lines *could* be visualized instead.

Nielsen points out the needs of aesthetic and minimalist design, which explains how "Dialogues should not contain information which is irrelevant or rarely needed" [46]. Combining our guideline with Nielsen's seems appropriate, to essentially not flood the screen with text messages just to replace color cues.

7.3.13 Provide easy access to help and documentation when needed

When handling complex software, confusion regarding certain functionality could potentially occur, the user should then have some way to learn about it. Constantly getting frustrated in the first phases of learning can be incredibly disheartening for the user, and following this guideline can at least alleviate that feeling. The main difference we made with our prototype to accommodate for this need is to add the tutorials to the start screen of Layout Designer. As many tutorials already exist but are currently accessed through a website that can not be reached directly through Layout Designer, to us it was an obvious improvement to incorporate them in a simpler way.

An aspect where we failed to implement easier access to documentation, and actually made it harder, was when we (accidentally) removed the "Whats This Tool"-button, that opens the documentation relevant to which object was pressed, an example of this can be seen in Figure 7.14. This button should simply be re-added as a new

button in the toolbar, and should be visible by default. We missed to add this "Whats This Tool"-button as there are two other help buttons, but these buttons are not linked to documentation, but just provides the user with ids in different ways.

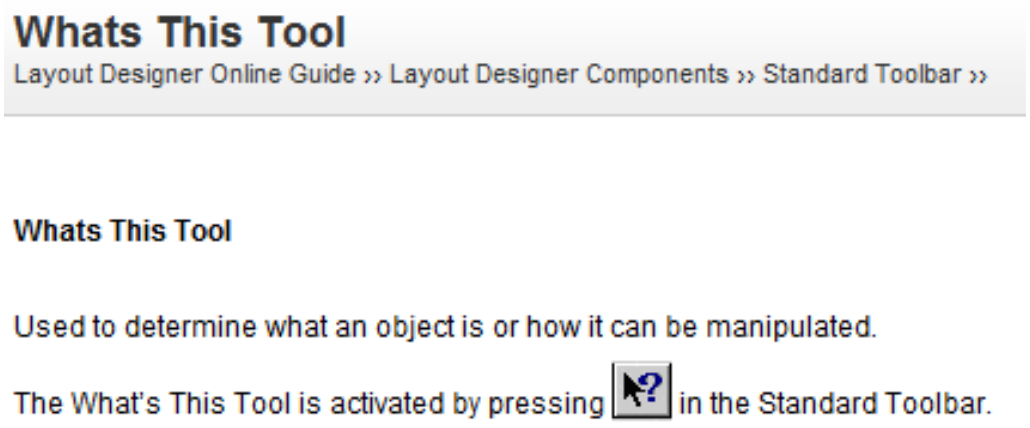


Figure 7.14: The documentation accessed by using the "Whats This Tool"-tool on itself.

This guideline does not really have an alternative, with the exception of having a software that is so easy to understand that neither help or documentation will never be needed. Nielsen [46] mentions this in his heuristic about help and documentation as well, that it is preferable if they are not needed but should be easily accessible when they are. In a functionality-heavy software we believe that documentation should exist as it will most likely be needed at some point.

7.3.14 Error messages should be clear, with connections to real life.

When coming up with this guideline, the main problem that we have attempted to address were the errors connected to the creation of new layouts (which was presented in Figure 6.2). The problem was primarily that the error output was not connected to the input in an intuitive way, as the error appeared several steps afterwards. This causes a lot of confusion as a first time user that does not remember what a *Drawing Name* is when presented with the error. There are similar problems with errors inside of the software that are not as common (which was presented in Figure 6.5). These errors appear when accidentally putting something outside of the workspace and then interacting with *any* other object, then the user has no idea what has gone wrong and will cause a lot of frustration.

Nielsen states that the system should have a connection to real life and talk the user's language, he also mentions that error messages should be expressed in a way the user can understand [46]. We interpret this as in order to make error or warning messages clear, they need to have a connection to real life. In real life, warnings

consistently uses bright colors, combined with either sounds or symbolism. While following our guideline it is valuable to keep two guidelines set by Nielsen in mind, which are “Match between system and the real world”, and “Error Prevention” [46]. When possible you should prevent the user from encountering errors as much as possible, but otherwise keep it oriented within the real world.

To be perfectly clear, this is not something that has been improved in our prototypes, but should have been. We have only given one example of how errors can appear, and that is in the start screen. In section 7.3.12 we have only given brief explanations of what is generally bad, and not given any concrete examples of solutions that would be good.

8

Discussion

During this project many things went well, and a few things did not go according to plan, or was not even planned for. Here follows a discussion about our process, our results, and potential future work.

8.1 Results

We are satisfied with what we have produced and presented as the result but we do acknowledge that there is still much that can be done. Feedback gotten during the final evaluation of our design gave insight about functionality that would need to be updated but that we have not worked with. In order to get a better and more complete design prototype we should have included more layout design users and should have done so earlier in the process.

The style convention is heavily based on Material Design [42], it worked well for us, and as material design is widely used it will work well for many others. We have made slight modifications to the guidelines given in material design to better fit our design but mostly it is just a concise presentation of the information we found most useful and important.

We feel like the guidelines will work for similar projects and the reasoning behind them has been described as part of the process to reach these guidelines. Some guidelines are, according to us, more important when designing complex software so even if they might be generic, the emphasis we have added is a contribution to answer the research questions. However some of the guidelines might be a bit too general and applicable on all design processes rather than specifically functionality heavy path drawing software.

8.2 Process

Overall we believe that our plan was well structured at the top level but it was lacking on the detail level and our execution of it was not flawless. To start of, we did probably spend a little too much time on the methodology in the prestudy, we wanted many different options to be prepared for different scenarios but most of

that work did not have any actual impact on our project.

During the prestudy we also spent some time getting to know the software, and we took notes on flaws the software had. This in itself was really helpful and when redesigning such a complex software it is probably essential. We did however not look back at our notes regarding this very often. Nearly all changes are done based on feedback from participants in our tests and interviews. It is of course good that the changes were made for a reason, but we might have been to cautious to not make changes on our own. The notes will most likely be appreciated by Kollmorgen to use in their future implementation of the new design and functionality, but it should have been used more during our process as well.

When it was time to start prototyping we got too excited, wanted to produce something that looked good immediately. We thought we were in agreement on how things should look and we started making prototypes in Figma, we quickly realized that our views varied quite much regarding what was of importance and how things should look. At this point we took one step back and did sketches together and then discussed the strengths and weaknesses of those, this made it possible to have a template of sorts for future designs. This is something we really should have started with from the beginning, the prototyping in Figma was much more efficient once we had sketches to base our prototypes on. Doing the sketches was also more efficient in determining how to divide the screen real estate as well as to ideate new ideas.

COVID-19 did not affect the overall project very much as the prototyping can be done just as well while working remote, and the usability testing can to an extent be done remotely as well. However, COVID-19 was of course not something we had planned for so it did affect certain aspects of our process, most importantly our goal to observe users of Layout Designer in their natural work environment. We spent time on planning how to perform our observations and talking with representatives at our company to plan visits. The planning took a long time so once we actually had participants to visit we would have to travel to Stockholm, and COVID-19 had started spreading in Sweden, mostly in Stockholm. Because of this we chose not to travel there. Since we were already behind schedule due to the time it took just to book the observations we wanted to get the users' input quickly so we decided to make a questionnaire while waiting for contact details to the users. Only four users of Layout Designer wanted to share their information with us and then only three answered, even after being reminded. This did not provide us with sufficient data to base any design decisions on. We do regret going for a half measure to try to save time here as the result was that we got nearly nothing at all out of the time we had already spent. It would probably have been more beneficial for us to try to book remote interviews with the participants, even if it was heavily delayed according to our initial plan. Another thing that potentially was negatively impacted by COVID-19 was our communication with the Kollmorgen team. We did not participate much in the team sprints, but we did participate in the stand up meetings every morning. Those meetings were discontinued by us when we started working from home and

we therefore lost part of our continuous communication.

For our final evaluation of our prototypes we would have wanted to have participants actually navigate through our designs, performing similar tasks to the ones we asked novice users perform in Layout Designer. This would have provided us with a better comparison regarding how much easier users experienced the navigation of the program. However, the evaluation we did instead did provide us with valuable insights regarding more detailed functionality and how certain tools are currently hard to work with.

8.3 Validity and Generalisability

We are confident in many of our improvements. Today there is a lot more horizontal screen real estate than vertical to use, removing the big toolbar at the top and placing a smaller one on the left side therefore feels appropriate.

The groupings we have made, regarding how to find similar tools and context sensitive information are valid solutions as well and can be backed up by the feedback we have received throughout the project. The novice users were trying to reach similar tools by interacting with one of the tools, i.e. the way you find other tools now. The novice users looked for related functionality close to the tool, not exactly how we have implemented it, but having a constant location for related functionality, as our design has, makes it apparent where to look and should therefore minimize the amount of confusion. Both of these ideas were also praised by the subject matter experts during our follow up interview. It is also much more similar to other very popular software, such as Adobe's Photoshop and Illustrator. The final version was also shown to users of Layout Designer and they said it was a good change, strengthening the validity for this change in this specific software. We believe that the wide usage of similar solutions and the wide acceptance of this concept is a good sign of re-usability, it could probably be implemented in a similar fashion in a majority of software with traits correlated to the ones found in Layout Designer today. However, the tools in the toolbar are now sorted by our perception of their relevancy, which might be incorrect because of our inexperience with Layout Designer, so this specific aspect of the toolbar not be as valid as the rest. We have designed how it can look when modifying the toolbar though, making only the default layout affected by our inexperience as each user can modify it to their own preference.

To interact with the settings of an object you have to navigate to the object's properties via a right click. We are convinced that our decision to move these settings, for the currently selected object, into the right side panel is a big overall improvement. This is a considerable decrease to excise when interacting with the settings. Since the work space is no longer occluded by the popup window you always get direct feedback when making changes. Also, with this design you can glance at the available settings much quicker and determine if the setting you want exists or not.

From our subject matter expert interviews we gathered that objects with the exact same configuration can be used in many different layouts. Our idea, inspired by Unity, to add the functionality to save objects as prefabricated objects, or "Prefabs" should help with this. Not only can one user reuse previous objects much easier between layouts, but in theory a library with common components could be made as well, which could help new users make better layouts if they have some components made by experts. Our idea is to select the items you want to save as a prefab and either drag those items into the prefab folder, or right click and select "save as prefab". We have not properly visualised the interaction on how to save and reuse these prefabs, which makes it impossible to claim that it is valid based on feedback. However we do believe that the idea is promising for other software where components can often be reused and we do believe it has some validity as the idea is used in one of the world's most popular game engines.

The CAD-layers we have come in contact with often had obscure names, so instead using an occluding modal popup without direct feedback to interact with the CAD-layers, it should be possible to immediately see what is part of the selected layer. The idea we had was a list of layers, available in the right side bar. This idea was well received both by users and non-users of Layout Designer. It is definitely an improvement as it no longer occludes part of the screen which presents relevant information for the current context.

8.3.1 Guidelines

The guidelines that were developed by us are all very valid for designing functionality-heavy CAD-drawing software's, but practically none of them are *only* applicable on that kind of software. This makes these guidelines usable for a lot more projects, but may be too generic to add any substantial value.

The guidelines that sticks out compared to the others are "Be mindful when occupying vertical screen space instead of horizontal screen space.", and "Understand core functionality before investing time in designing for it.". These are particularly worth thinking about when designing for a screen space which generates value for the user as the work space gets larger. But even these guidelines are not specifically exclusive for the CAD-drawing software we had in mind.

Some guidelines might be too general and feel superfluous at a glance. However, we do believe that they are still of value for this domain. Some of them are general, there is no denying that, but we believe they might be more relevant than other general guidelines are for designing a functionality-heavy software.

8.3.2 Style conventions

The style conventions for this project were heavily, almost entirely, based on guidelines from material design. These worked quite well for us and since material design is already widely used we are certain that style conventions of this character can

be used for similar projects as well. Some modifications were made to better fit into our user interface design at points, but the site material.io does provide limited instructions, perhaps better expressed as examples, on how to adapt their base components to better fit into the theme of your software. So while the style conventions we made are pretty close to the default material guidelines, we do believe they can be used in a wide selection of similar software with just minor modifications, i.e. they are not general as is, but close to.

8.4 Ethical aspects

In order to find an answer to the research question we have depended on interviews, observations, and usability tests. We have put emphasis on making the participants clearly feel that it is the product that is being tested and not them. Also, that no information from them has been included in the report without their approval.

An important change that we have wanted to make for the users that regularly use Layout Designer was to improve their working conditions as much as we could, and focus on the user's needs as much as possible. We would like to believe that we have thought of the users in each step of our process, and each little aspect has them in mind to some extent.

The main disability that we have thought about while developing our prototypes are color blindness. We have explored different color schemes, and which scenarios could be troublesome for the user. One of these troublesome scenarios are the errors that occasionally occur, as the red tint that is shown could appear as a light gray instead, which is why it is accompanied by the recognizable exclamation mark. An example of this error handling is shown in Figure 7.7 (g).

8.5 Future work

We believe Appendix B, which consists of notes regarding potential issues in Layout Designer, to be a valuable resource in the continued work of this product. We have taken notes of issues we encountered while interacting with Layout Designer. The issues are part of both the user experience and the user interface domains. Certain items of the list are not big concerns, there are simple things such as limiting movement until a button is pressed. In such a case that might be the desired behaviour in order to minimize the size of errors if something is accidentally moved, but if it has been noted by us it was something that was not apparent and might need some clarification. Some of the issues have been solved, to an extent, in our current designs. An area of our designs that currently have some inconsistencies are the checkboxes, we suggest having the checkboxes to the left, and then have the text following the checkbox. However, in the sidebar we believe that it is worth considering to deviate from the convention. Shown in Figure 8.1 are both checkbox positions, in which we believe the consistency of interactive elements' position in Figure 8.1 (b) is more important than following the convention of having the checkboxes to the left as shown

in Figure 8.1 (a).

(a) **(b)**

Figure 8.1: Checkbox position alternatives in sidebar.

An area we have completely overlooked is the interaction with the cluster functionality. According to the participants in the last evaluation there are many different alternatives one can choose when using the cluster tool, many of which are not apparent regarding their functionality.

A keybind to allow interaction *through* the CAD-drawing, and light up in the CAD-layer tab should be available. A particular keybind for this actions has not been evaluated yet, and would be usable according to feedback given from several sources at the company.

8.5.1 Implementation

A rather apparent approach to continuing this work is implementation. As of now everything is theoretical, while it does seem realistic we can not know if some parts of the design will be troublesome until it is actually realised into a functional prototype or a fully functioning product.

There are also certain aspects that are easy to overlook when you are not actually using the software yet. We have started thinking about key bindings but without actually testing them in the natural flow of working it is hard to know if they feel natural or not.

8.5.2 Evaluation

More evaluation of the designs would be needed before deploying the product. Especially evaluation of a functioning prototype to be able to better assess how users

would interact with the new design.

Although we have narrowed down the prototypes to essentially two different color palettes, we did not receive enough feedback during the last evaluation regarding these to make a definitive decision with users in mind.

8.5.3 Further Redesign

Although we like how our prototype looks a complete redesign of all the displayed colors in the software would have to include more usage of an accessibility tool to make sure colors are friendly to colorblind users, and in the cases when this can not be accomplished, make sure to convey information not only with colors.

Currently the parts of splines that will not function, due to e.g. a too tight curve, is only shown with a color. The user can also choose to use different colors for splines with different settings. In theory this means that the error color can be exactly the same hexadecimal color code as one of the spline types, or that even different color codes can be perceived as the same by anyone who is colorblind. A way to implicitly express erroneous areas of a spline visually in a non color dependent manner has to be explored.

9

Conclusion

During this project it has been researched how to redesign an old software and in doing so what to consider when designing for a functionality-heavy path drawing software. The results are interesting because of their potential use for future designing and redesigning of complex software. The project was planned to take place mainly at Kollmorgen Automation AB in Mölndal, but was, due to COVID-19, executed remotely to a big extent. Our concept is a modernized and improved version of Layout Designer. It has better grouping of functionality and easier access to settings. It decreases the amount of excise by removing unnecessary popup windows. The concept is purely visual at this point, nothing has been implemented. The project was done in iterations, prototyping ideas, evaluating the ideas, and compiling guidelines based on the experiences.

The **process** related guidelines are

- Continuous and documented communication is key.
- Confirm common goal between stakeholders and designers.
- Have a detailed plan of how to handle each screen.
- Schedule third party reliant activities long before they have to be carried out.
- Structure the design workspace and group related concepts.
- Have a template of your design that showcases only the essential parts.
- Create style conventions for your project early on.
- Confirm your understanding of core functionality before investing time in designing for it.

The **design** related guidelines are

- Provide visual feedback related to the software's status when relevant.
- Group information and functionality in a way that makes the user see the correlation between relevant elements
- Be mindful when occupying vertical screen space instead of horizontal screen space.
- Try to convey important information with other cues than color, especially when errors may have real world consequences.
- Provide easy access to help and documentation when needed.
- Error messages should be clear, with connections to real life.

To continue the work of this project a functioning prototype would have to be developed. This is especially important to be able to test the designs in a natural work environment and give an accurate estimation regarding how productivity has been affected by the design changes. A few issues with Layout Designer were brought up in the last evaluation and have not been addressed before the product could be considered finished.

Bibliography

[1] "Phase two of the gothenburg project has just been wrapped up!," 2019. https://www.letsholo.com/gothenburg Accessed 2020-01-28.

[2] M. Ansaldo, "How a robot vacuum navigates your home," 2018. https://www.techhive.com/article/3281014/how-a-robot-vacuum-navigates-your-home.html Accessed 2020-01-28.

[3] J. Tidwell, C. Brewer, and A. Valencia-Brooks, *Designing Interfaces : Patterns for Effective Interaction Design*. O'Reilly Media, Incorporated, 2020.

[4] "The outbreak of coronavirus." https://www.chalmers.se/en/news/corona-virus/Pages/default.aspx Accessed 05 Jun 2020.

[5] "Agv controls solution." https://www.kollmorgen.com/en-us/products/vehicle-controls/ Accessed 2020-01-31.

[6] J. Nielsen, "10 usability heuristics for user interface design," 1994. https://www.nngroup.com/articles/ten-usability-heuristics/ Accessed 2020-01-31.

[7] A. Cooper, R. Reimann, D. Cronin, and C. Noessel, *About Face*. John Wiley & Sons, Inc, 2014.

[8] "Aethon. (2020). Aethon - Autonomous Mobile Robots - Industrial, Healthcare and Hotels.." https://aethon.com/ Accessed 12 Feb. 2020.

[9] "Cobalt Robotics | Security Robots at Work." https://cobaltrobotics.com/ Accessed 12 Feb. 2020.

[10] "Fellow AI | Supply Chain Automation." https://www.fellowai.com// Accessed 12 Feb. 2020.

[11] "Kollmorgen Corporation Celebrates 100 Years of Innovation." https://www.kollmorgen.com/en-us/company/history/ Accessed 06 Mar. 2020.

[12] "Why Motion Matters." https://www.kollmorgen.com/en-us/company/why-motion-matters/ Accessed 06 Mar. 2020.

[13] "Buy adobe photoshop | best photo, image, and design editing software." https://www.adobe.com/products/photoshop.html Accessed 2020-02-05.

[14] A. K. Bedakaa and C.-Y. Lina, "CAD-based robot path planning and simulation using OPEN CASCADE," in *Procedia Computer Science, volume 133*, ACM Press, 2018.

[15] G. Lee, CharlesM.Eastman, TarangTaunk, and Chun-HengHo, "Usability principles and best practices for the user interface design of complex 3d architectural design and engineering tools," in *International Journal of Human-Computer Studies*, pp. 15 pp.–90, January-February 2010.

[16] I. Pettersson and W. Ju, "Design techniques for exploring automotive interaction in the drive towards automation," in *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, (New York, NY, USA), p. 147–160, Association for Computing Machinery, 2017.

[17] P. Neto, J. N. Pires, and A. P. Moreira, "Robot path simulation: a low cost solution based on cad," *2010 IEEE Conference on Robotics, Automation and Mechatronics*, 2007.

[18] R. S. Dicks, "Mis-Usability: On the Uses and Misuses of Usability Testing," in *Proceedings of the 20th annual international conference on Computer documentation*, ACM, 2002.

[19] J. S. Dumas and J. C. Redish, *A practical guide to usability testing.* Intellect Books, 1999.

[20] F. Dias and A. C. R. Paiva, "Pattern-Based Usability Testing," in *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE, 2017.

[21] J. Tidwell, *Designing Interfaces : Patterns for Effective Interaction Design.* O'Reilly Media, Incorporated, 2011.

[22] "Microsoft Excel." https://www.microsoft.com/en-us/microsoft-365/excel Accessed 18 May. 2020.

[23] J. Zimmerman, J. Forlizzi, and S. Evenson, "Research Through Design as a Method for Interaction Design Research in HCI," 2007. Human-Computer Interaction Institute. Paper 41. http://repository.cmu.edu/hcii/41.

[24] W. Gaver and William, "What should we expect from research through de-

sign?," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, (New York, New York, USA), p. 937, ACM Press, 2012.

[25] H. W. J. Rittel and M. M. Webber, *Developments in Design Methodology*. John Wiley & Sons, 1984. https://cec.prodwebb.lu.se/sites/cec.prodwebb.lu.se/files/rittel_and_webber_1973_planning_problems_are_wicked_problems.pdf.

[26] B. Hanington and B. Martin, *Universal Methods of Design - 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions*. Rockport Publishers, 2012.

[27] Y. Wadsworth, *Do it yourself social research*. Left Coast Press, 2011.

[28] J. Nielsen, "How Big is the Difference Between Websites?," 2004. https://www.nngroup.com/articles/website-usability-differences/ Accessed 11-02-2020.

[29] B. Hanington and B. Martin, *Universal Methods of Design - 125 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions - expanded and revised*. Rockport Publishers, 2019.

[30] "How Big is the Difference Between Websites?." https://www.surveyking.com/help/likert-scale-example Accessed 12-02-2020.

[31] W. C. Adams, *Conducting Semi-Structured Interviews*, ch. 19, pp. 492–505. John Wiley & Sons, Ltd, 2015.

[32] "Figma - The modern interface design tool." https://www.figma.com/ui-design-tool/ Accessed 19 May. 2020.

[33] "OBS: Open Broadcaster Software." https://obsproject.com/ Accessed 3 Mar. 2020.

[34] "The state of the art of illustration.." https://www.adobe.com/products/illustrator.html Accessed 2020-05-19.

[35] J. Pot, "Wait, is dark mode actually bad for productivity?," 2019. https://zapier.com/blog/dark-mode-bad-productivity/ Accessed 06 Mar. 2020.

[36] "Coloors - the super fast color schemes generator!." https://coolors.co/ Accessed 2020-05-19.

[37] "Colormind - the ai powered color palette generator." http://colormind.io/ Accessed 2020-05-19.

[38] "Palettte app." https://palettte.app/ Accessed 2020-05-19.

[39] "Paletton - the color scheme designer." https://paletton.com/ Accessed 2020-05-19.

[40] D. Nichols, "Coloring for Colorblindness." https://davidmathlogic.com/colorblind/#%23000000-%23121212-%23383838-%230078D7-%23CF6679 Accessed 20 May 2020.

[41] "Color tool - material design." https://material.io/resources/color/#!/?view.left=0&view.right=0 Accessed 2020-05-19.

[42] Google, "Material Design - Design." https://material.io/design Accessed 18 May 2020.

[43] "Research." https://material.io/components/text-fields#research Accessed 06 Jun 2020.

[44] "Chat, meetings, calling, collaboration | microsoft teams." https://www.microsoft.com/en/microsoft-365/microsoft-teams/group-chat-software Accessed 2020-05-19.

[45] D. Flück, "Protanopia – Red-Green Color Blindness." https://www.color-blindness.com/protanopia-red-green-color-blindness/ Accessed 21 May 2020.

[46] J. Nielsen, "10 Usability Heuristics for User Interface Design." https://www.nngroup.com/articles/ten-usability-heuristics/ Accessed 21 May 2020.

[47] B. Tognazzini, "First Principles of Interaction Design (Revised & Expanded)." https://asktog.com/atc/principles-of-interaction-design/ Accessed 21 May 2020.

[48] "Home | trello." https://trello.com/ Accessed 2020-05-20.

A

SME Interview Template

Date:
Interviewee:
Interviewer(s): Adam Kjellgren & Erik Hildinge.

Hey, We're Adam & Erik and we're interaction design students at Chalmers. We're currently evaluating Layout Designer. We are going to ask you to perform some tasks, while we observe. We want to emphasize that we are evaluating Layout Designer, we are not evaluating you. It is always okay to refuse to answer any question if you don't feel comfortable answering it.

We would like to record this as it makes it easier for us to double check that what we remember from the discussion is actually correct, as well as it makes it easier for us to be active participants in the discussion if we don't have to make perfect notes while we talk. It will also be useful for us if we want to quote you, if we want to do that we will of course get in touch with you with the specific quote and ask if it's okay, which you can of course say no to. We'd like to record the screen as well as it will make it easier for us to follow your course of action when we review this later.
All information will be presented as "Interviewee X" or similar, in the report.
Okay with voice being recorded?:
Okay with screen being recorded?:

Warm up (5 minutes)
How did you get into this field?

- Creating layouts in a CAD based program, or more specifically Layout Designer?

How long have you been using Layout Designer?

- Are you still actively using it?
    - How long ago did you stop?

Have you been involved in any development of Layout Designer?

In what capacity have you used Layout Designer?

- Taught others, only here in the office, or on site making layouts either on your own or together with someone else?

Do you know if it is common that clients need help to modify or create their layouts?

- How and when? Or maybe more importantly, why do you think that is?

How has Layout Designer changed in the past few years?

Interview (20 minutes, in front of the software)
<u>Showcase</u>
We want you to give a brief introduction to Layout Designer. Imagine that we are total beginners, what are the first things that you would show us?

We want you to create a path around the warehouse, with one starting station, and one end station.

- Obviously that was a quick demonstration, but do you have an estimate on how long it takes to make an actual functioning layout?

Are there any keybinds/short-commands in Layout Designer that everyone should know about?

Questions

What are the most essential functionality or tools in Layout Designer? Could you point them out and describe what you use them for? (alternatively: What tools could you not be able to work without?)

- Is there anything that could improve these for you?

Is there any functionality that you rarely use?

- Why is this functionality rarely used and what could be done to improve it according to you?

What functionality or tools that you know of, have you never used?

- Do you know of an instance where it could be useful?

Is the manual reflector tool ever used or do you always import their positions from a text file?

What are the most critical changes that must be performed to Layout Designer to keep it competitive in the future?

What is the best resource for people who want to dive in deeper? How did you learn the most efficiently?

What is one piece of practical advice you would give to someone starting out?

Wrap-up

Is there anything else you'd like to express for us to consider in evaluation of Layout Designer?

Thank you very much for your participation. You've helped us a lot! Would you be interested in a follow up interview after we've performed a couple of iterations of redesigning?

Let us know if you're interested in the results and we can send you a link when the report is published.

B

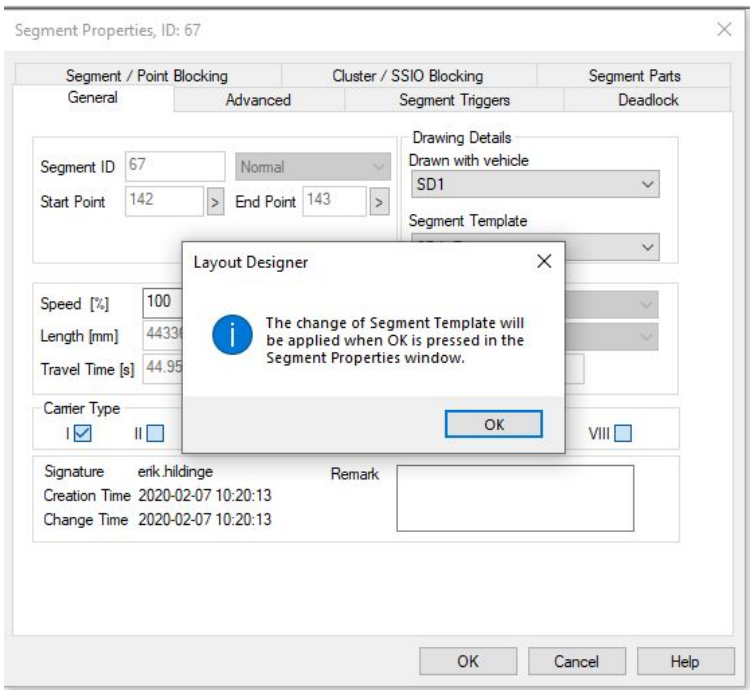
UI and UX issues in Layout Designer

Startup

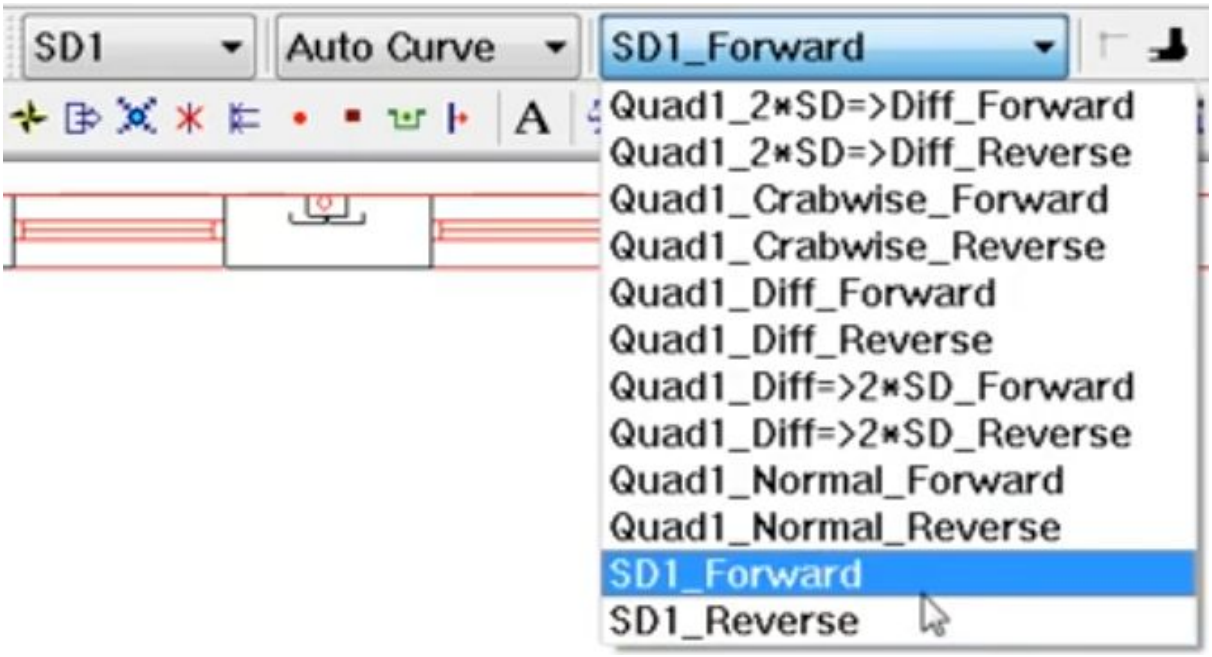
- “Drawing name” is unclear to us, as new users this gives us the impression that we can name the drawing ourselves and by doing so creating a new one, not that we have to know the exact name/filepath of an already existing drawing.
    - Should probably be named “Filepath for drawing” or simply not offer a textfield and just the browse option to select the existing drawing.
    - Nothing states that you actually need to have an already existing DWG file to use as the background.
    - A modal popup simply states “The drawing ‘X’ could not be loaded!” when trying to start the program, this should not appear at all but if it does it should appear already when trying to choose this drawing, not several steps later.
- Might be superclear when you have an actual vehicle as reference but realistic measurements for wheel base etc is hard to know when simply trying to learn the program.
- We’ve only come across “FREE NAVIGATION” as of now, there are no other options in the dropdown for “type of system?”.
    - Is free navigation the only option always, should it just be presented as information, or completely excluded as it is always the same and the information adds little to nothing of value for the user?
- If you deselect all carrier types it instead automatically selects all of them. Weird interaction that we’ve never seen before.
    - Unclear that carrier types are just your different modifications of different vehicles, makes it look like your vehicle has to fit into a certain type of carrier, not that you’ve chosen to make a new version of a carrier.
- **Vehicle setup**
    - Many options (max steering angle, speed, acceleration, sideway acceleration, max steering angle speed) are grayed out, indicating they can’t be changed, but are actually part of relevant settings and interactable by pressing on them.
    - Instead of giving information about the units (Eg cm/s, m/s, degrees, etc) to be used in the different input fields directly in the fields (or with a eg tooltip) the user has to find this information in the help menu.
    - Modal popup when adding a new vehicle to warn about effects of changing values later, too much excise.
    - Instead of not activating the button to proceed until the vehicle has all relevant information modal pop-ups are used to warn you about it when you try to proceed, as usual, unnecessary excise.

Spline tool

- When editing properties in a spline, a modal popup shows up. When trying to change something in the modal popup, a new modal popup was shown.

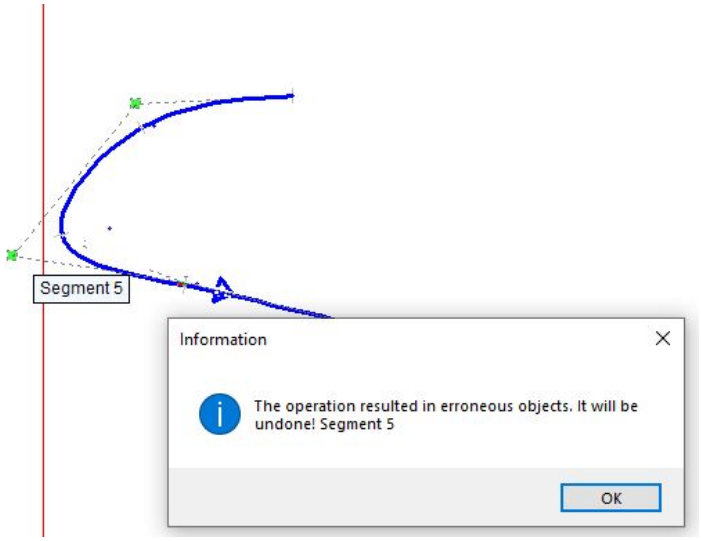


- When choosing a template for the vehicle, the Quad vehicle templates are visible when currently selecting a SD1 vehicle.



- When creating a spline, and you want to remove it, pressing delete only removes the path, but not the end points.
- When creating a spline curve, if the cursor moves outside of the area when holding down the mouse button (even if the user does not release the mouse key), the spline curve will be removed.
- After each spline you get an optimization report in a modal popup. This can be deactivated, but if you want this information you have to deal with a popup each time, that is not good, this should be able to position somewhere else.

    - The information in this optimization report is unclear to us as novice users, is this information ever relevant?
    - What was optimized is not shown. A before and after **should** be displayed.
- When editing a spline it automatically undoes changes if the first one isn't correct, instead of letting you perform multiple edits to position it correctly.
- Splines that are actually working and inside the layout will result in erroneous objects just because one of the points used to create the spline is outside the layout boundaries.



Line tool

- When trying to move to points (that are connected to the line tool) in a weird way, the user gets interrupted with a modal popup that states that the the action resulted in an erroneous object and the action will be undone. This is super inconsiderate since users don't do all actions at once.

Blocking detection

- To enable the detection of solid objects, the user needs to go into "Setup -> Auto Blockings" and then enable the checkbox "Detect Solid Objects". To then detect the collision areas, the user needs to go into "Tools -> Generate Blockings / Solid Detect" and do a scan. If the dimensions of the vehicle is too big for the path to fit, this will show now.
    - The european standard is that the vehicle is never allowed to be closer than 50cm to walls and other obstacles. There is no clear option to show this without going into the "mm" tool and measuring, or adding 50cm extra to the vehicles dimensions.
    - To have this important feature so hidden away and gaten between two different toolbar navigations seems like a time waste.

Navigation radius

- Is this used very often? It is added in a very complicated way at the moment.

Reflector lock

- This is set by default. Making the interactions with reflectors weird for us as novice users as they can't be deleted because of this and nothing indicates that it is because of this option.
- Reflector lock (among other functionalities can be changed from the view settings, very unintuitive as these do not affect the view, open extra windows or such, they are just settings regarding how the program acts.

Spots

- Not clear what the actual usage of these are. What do they do, when are they needed, how should they be used?

Navigation Wire

- Can snap to 0/90° and be unable to change back to the correct angle.
- When entering NW setup the default selection is "OK", so if you write the radius you want and press enter, it closes the dialog without saving the new value, unknowingly to the user.
- It states 15° increments but they are 45° increments.

Points

- To be able to move points in any other way than in the direction of the point, "SPACEBAR" needs to be pressed

Arcs

- Can only be moved along the initial (no longer visible) line. This does not follow my mental model well at all.

Selection

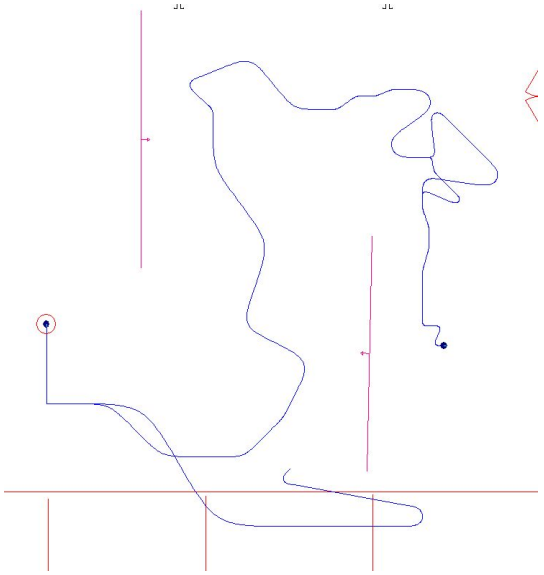
- This is inconsistent.
    - Sometimes hollow objects can be selected by pressing inside, other times you have to press exactly on a line or point.

Modifying drawing

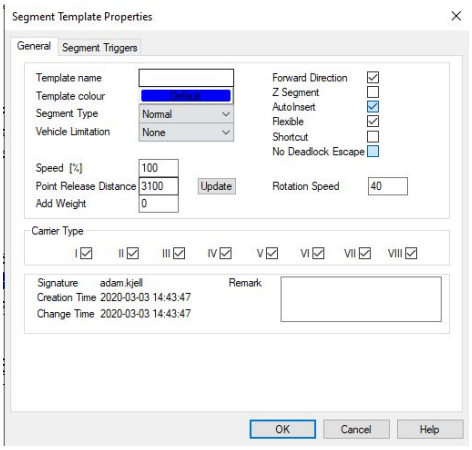
- Names of different parts of the drawing are unclear, so you don't know what they represent, to add onto this, your selection is not shown in anyway (in the layout) until after you've performed an action. Hiding or showing a layer is therefore an iterative process where you just have to try one at a time until you find the one you want.

Opposite direction segment

- After creating an opposite direction segment in the popup it can be continuously pressed to create new ones. However, new ones aren't created, you're just given information that 0 has been created, 0 exists (but it does since we just created the first one), and 0 has failed.
- These segments are on top of eachother, if they are both selected by dragging over them and then trying to move, change for one of the segments can be undone because of an error while visually, the other one is changed.



Segment

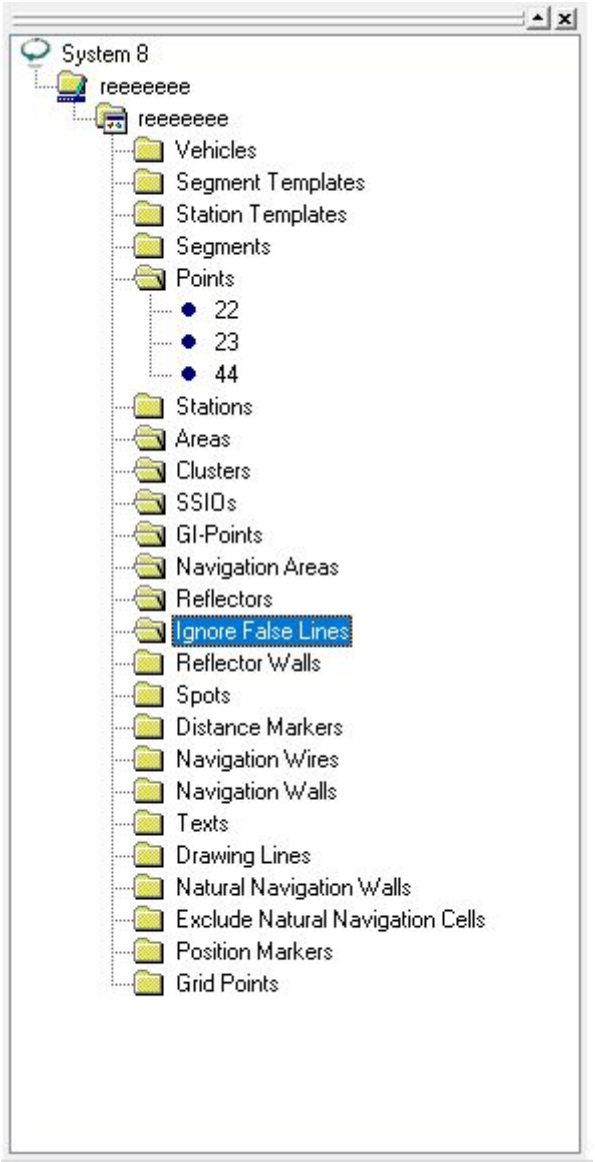
- When creating a segment template there are a few issues.
  - The segments can have backwards direction, but there is no option for it, only an option to deselect the forward direction.
  - 

  - The color of the text inside the template color is hard to see because of bad contrast. Is the text even needed?
  - You can accidentally deselect a checkbox by pressing the text next to it.

- The tooltips only say “Object level” and “Default level” for all options instead of telling the user anything about the effect each option has.

Layout Control

- Is it really needed to have all the folders present from start if most of them are empty? Seems like unnecessary excise, especially when they aren’t in alphabetical order (and no way to change that?).



Miscellaneous bugs

- It is possible that the program stops showing which tool is selected and which settings that are active. Makes it hard to work with and requires a restart AFAIK.
- If you place too many splinepoints a modal tells you that the max is 40, instead of just ignoring the last input it deletes all your progress.
    - This max is also incorrectly programmed, it says the max is 40 but it is when placing the 39th splinepoint that the modal shows up and deletes everything.
        - so even if you know about the “max 40” it can delete your progress because of incorrect information.
- Error modals aren’t always on top of the program, so if minimized and then opened from the taskbar the program can appear frozen because a currently hidden error warning prevents all interaction.
- When two spline points get placed outside of the layout because of the optimizing: If you drag one of the points inside of the layout **while holding shift**, the program always crashes.
    - If a splinepoint gets outside of the map, **nothing** is possible to move without getting an error.
- When going into “Setup -> Drawing” and clicking a checkbox: if you click anywhere else to deselect the checkbox, it reverts your change.

General about the program

- The values at the bottom are sometimes hard to associate with anything. Might need some clarification about what they represent in order to offer anything of value to the user.
- **Inconsiderate** - The layout does not remember which layout the user most recently used. It should.
- “Ctrl+z” or “undo” does not work after the user creates a path. I should be possible.
- The program activated certain options automatically. Uncertain if it should be like that.
    - If the user disables options in the toolbar, the program does not remember those choices.
- Even though we are designing for new users, nobody wants to remain a beginner. About Face p.239
- Can’t delete things if they are grouped
    - Delete still available in the right-click menu, should be deactivated when not actually available.

C

Large images for greater detail

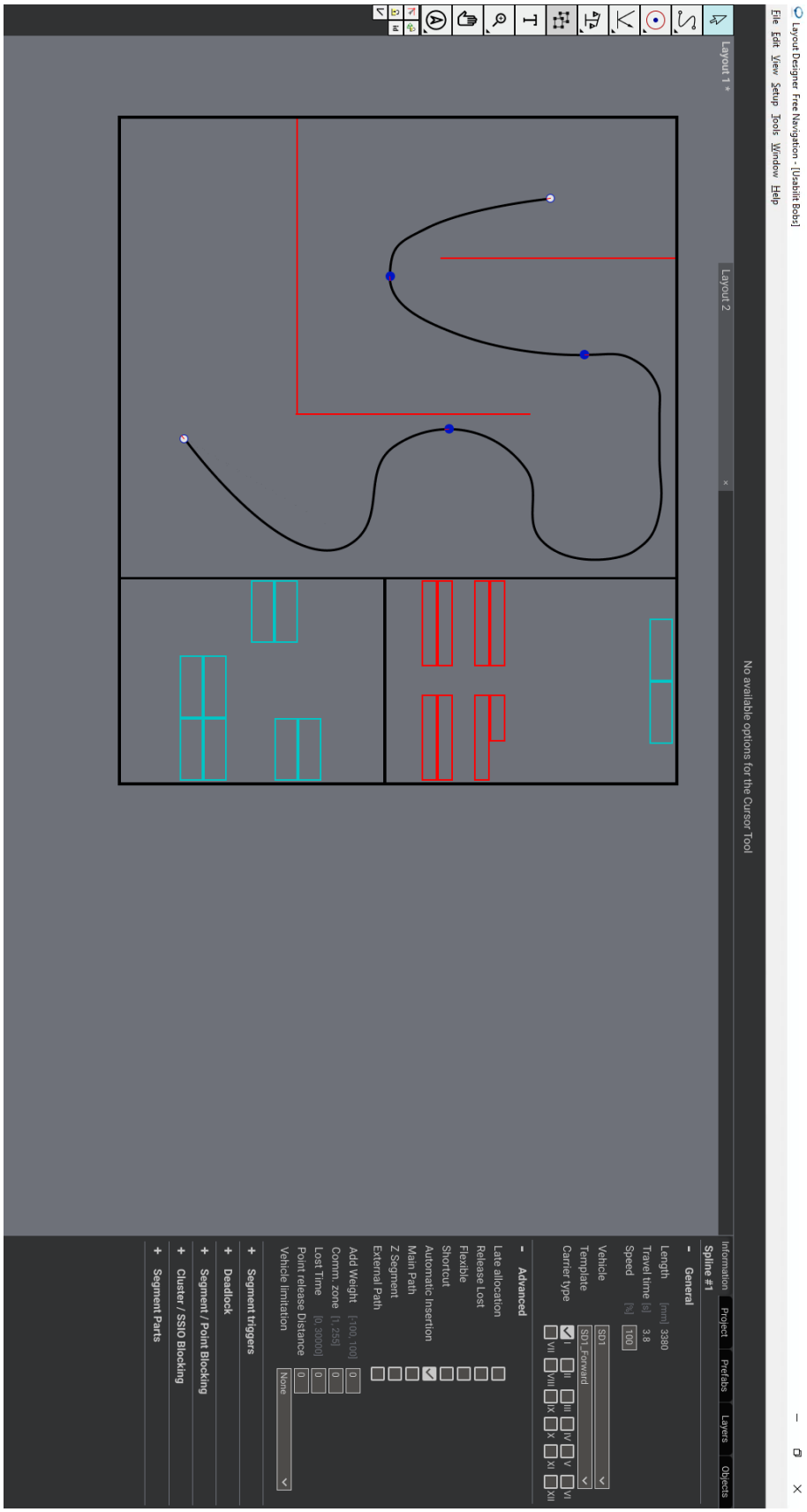


Figure C.1: The work space visible with an open layout and a spline currently selected.

Figure C.2: The start screen with the tutorials turned off.

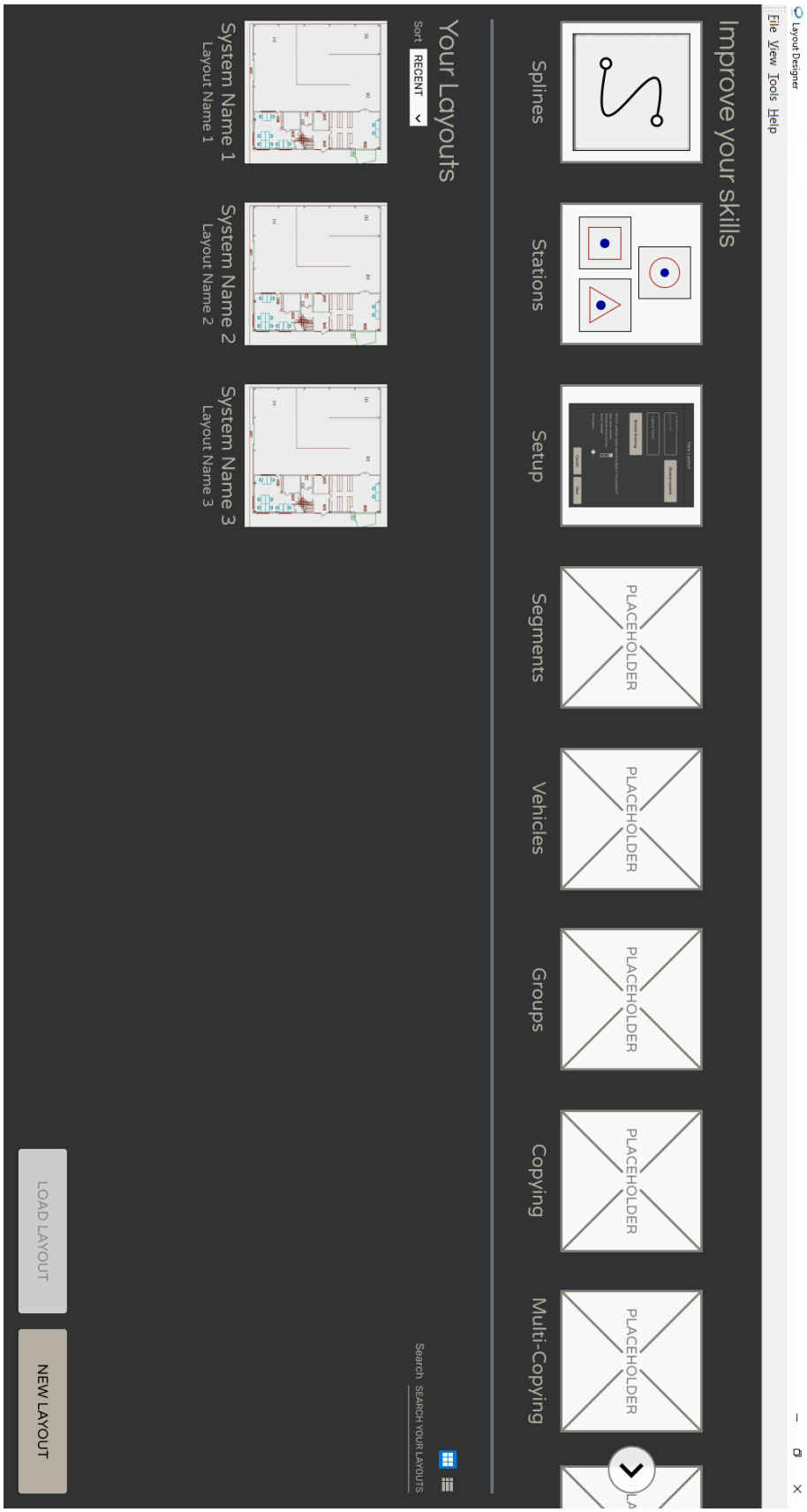


Figure C.3: The start screen with the tutorials turned on.

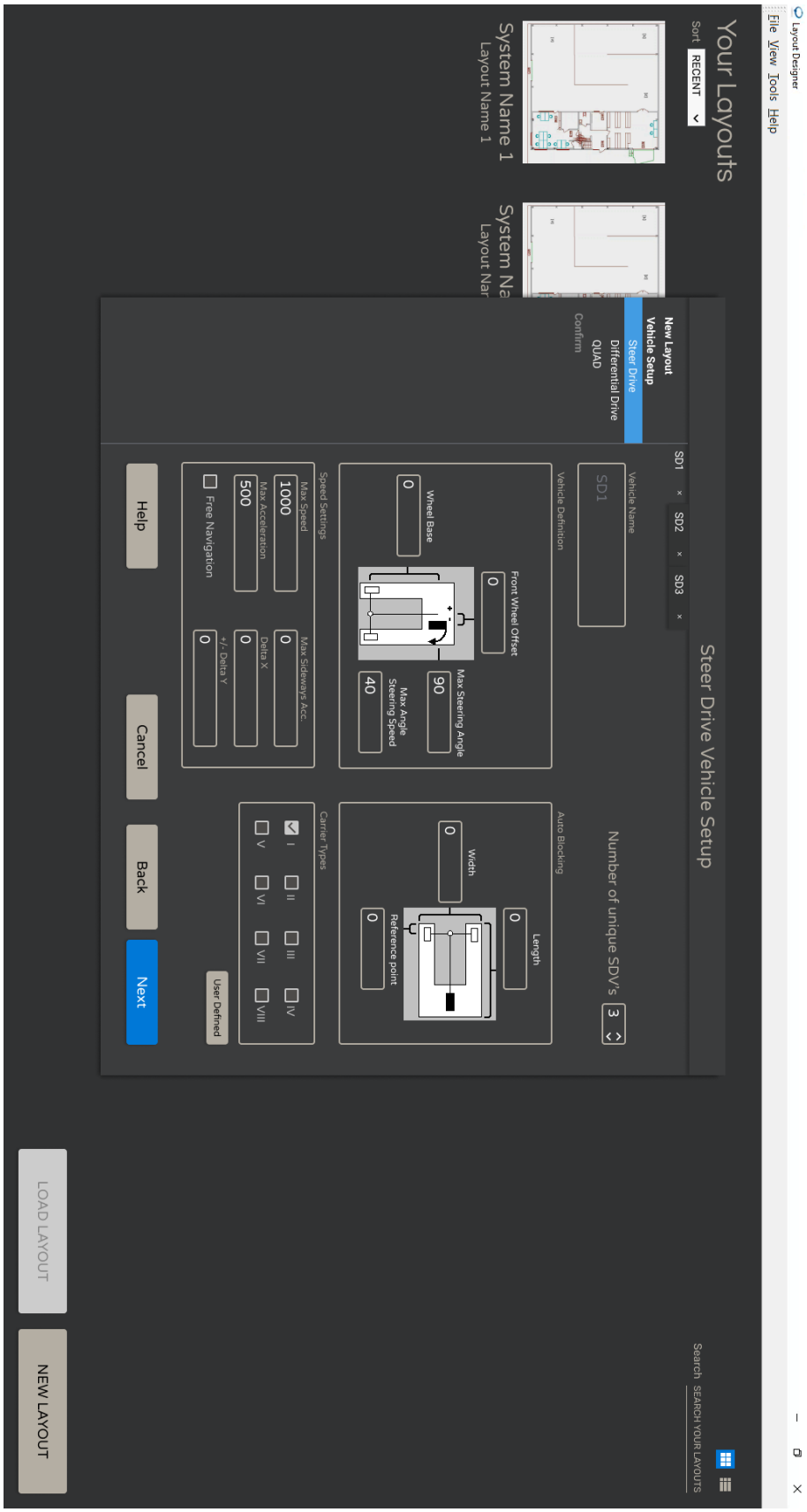


Figure C.4: Overview of the new layout setup window

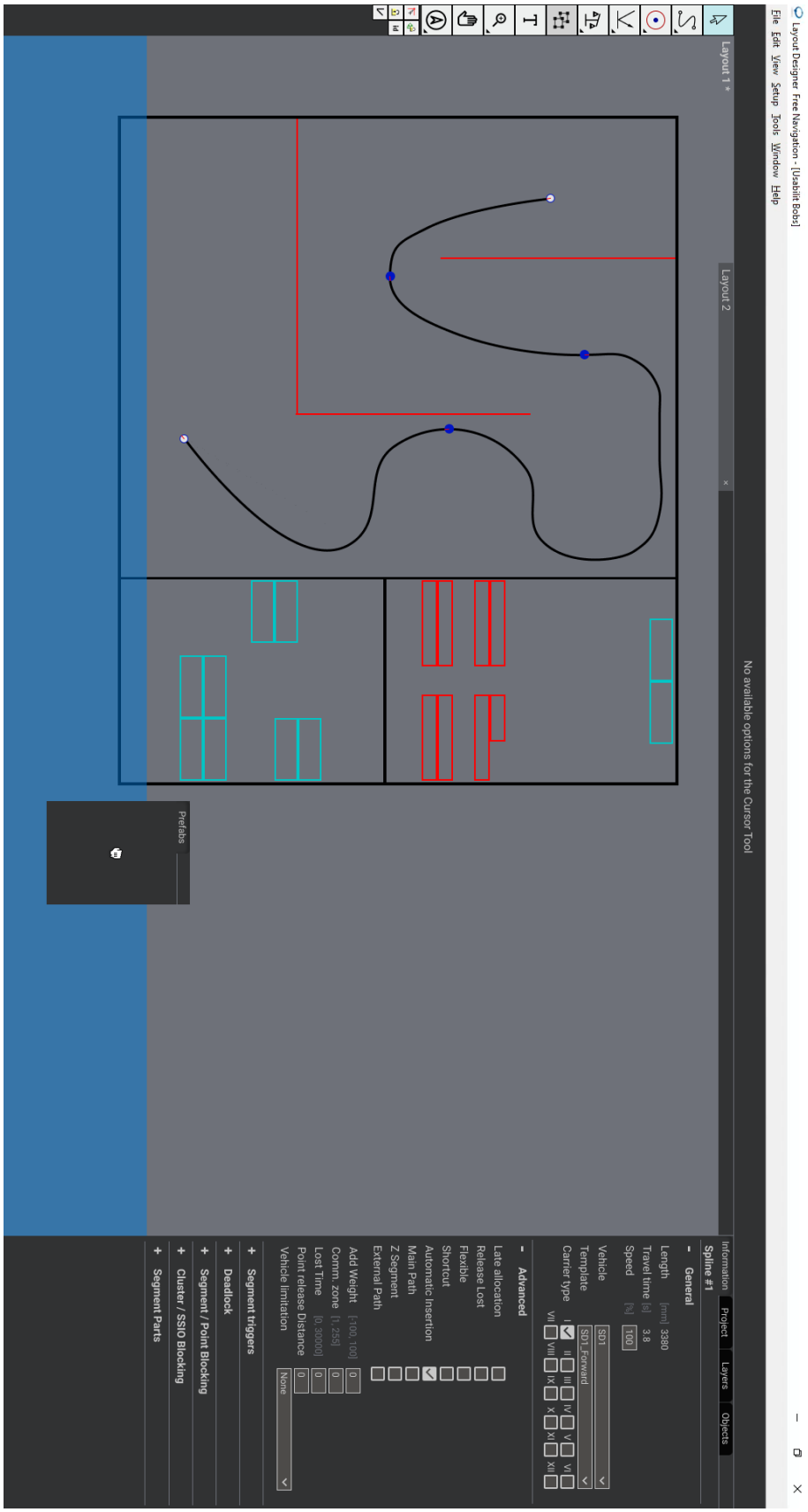


Figure C.5: Visualization to indicate where a window will be placed when moved by the user.

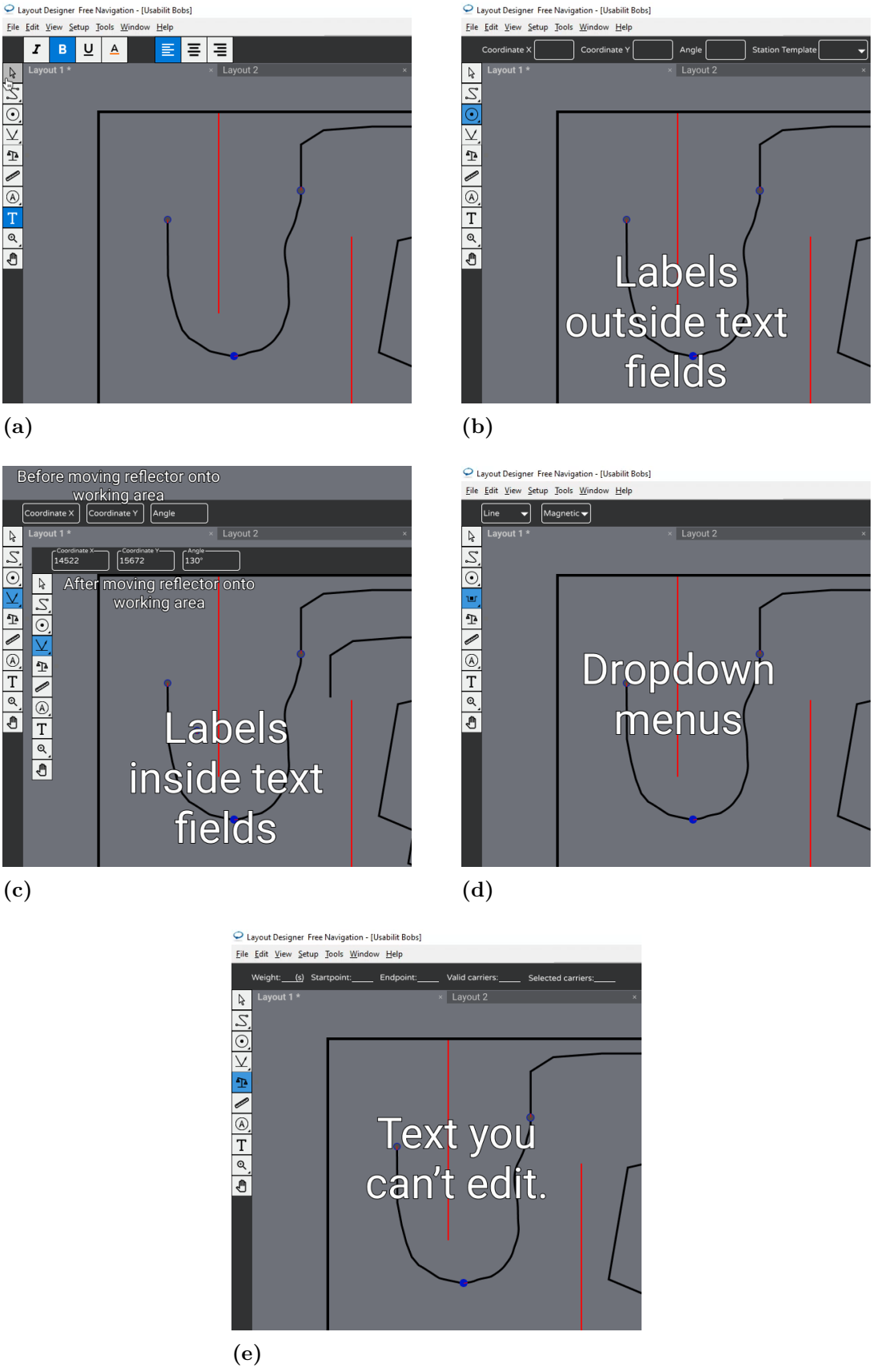


Figure C.6: Examples of what information to show in the context sensitive upper bar, with added suggestions to explore further.

Figure C.7: The default layout of the toolbar in Layout Designer

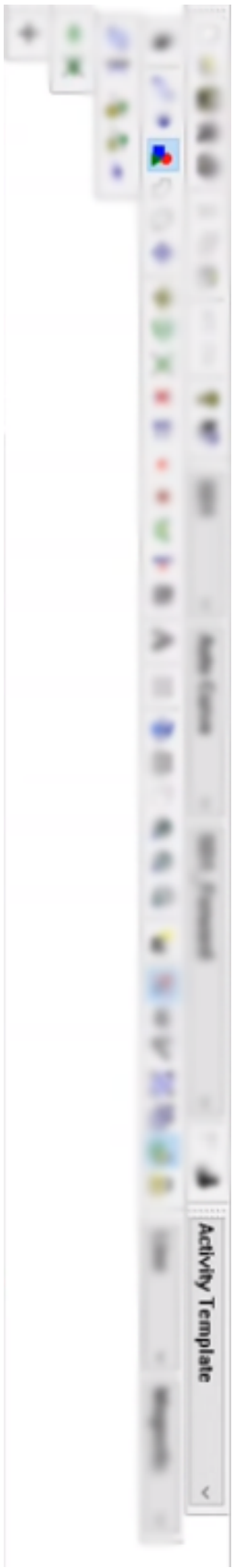


Figure C.8: Selected station tool and corresponding drop down menu in focus.

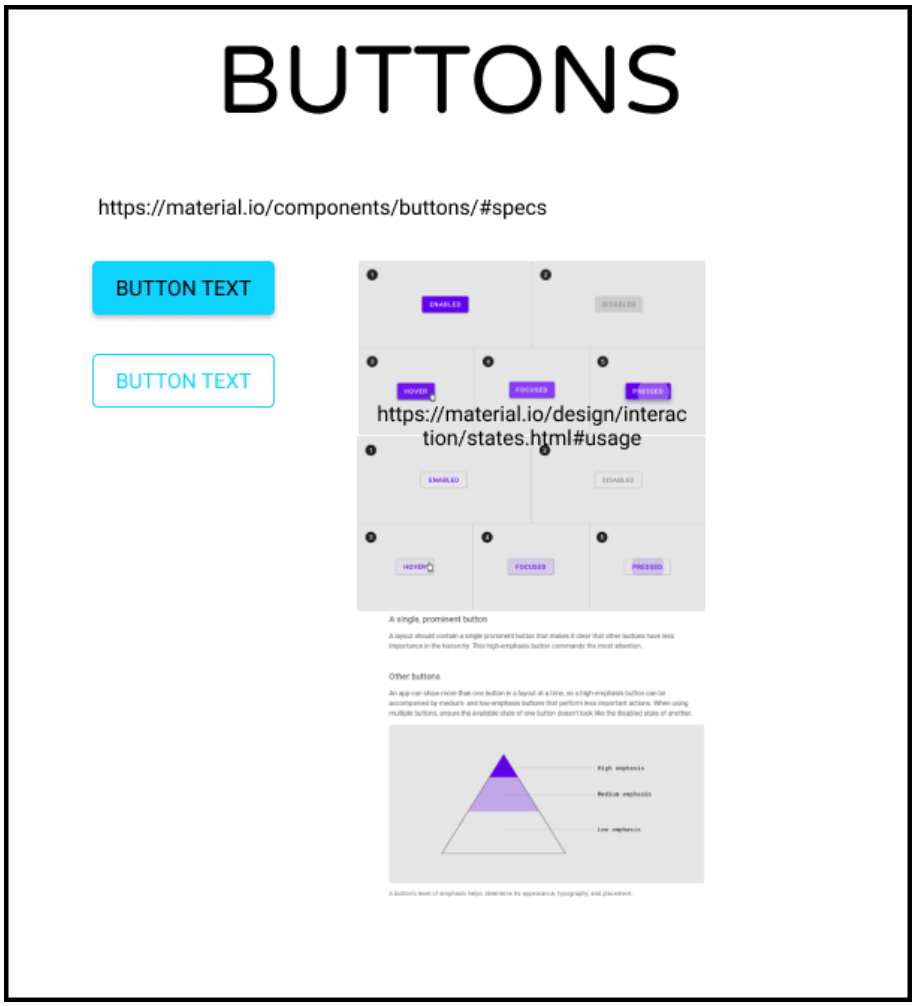


Figure C.9: Material Design based style convention for buttons

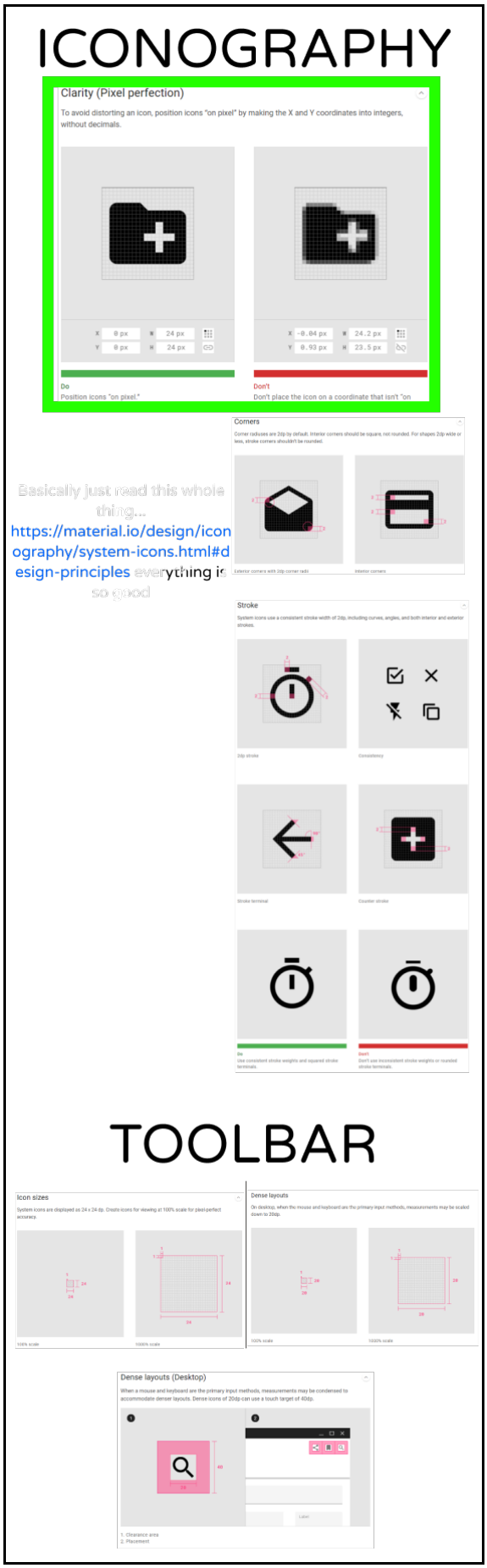


Figure C.10: Material Design based style convention for icons

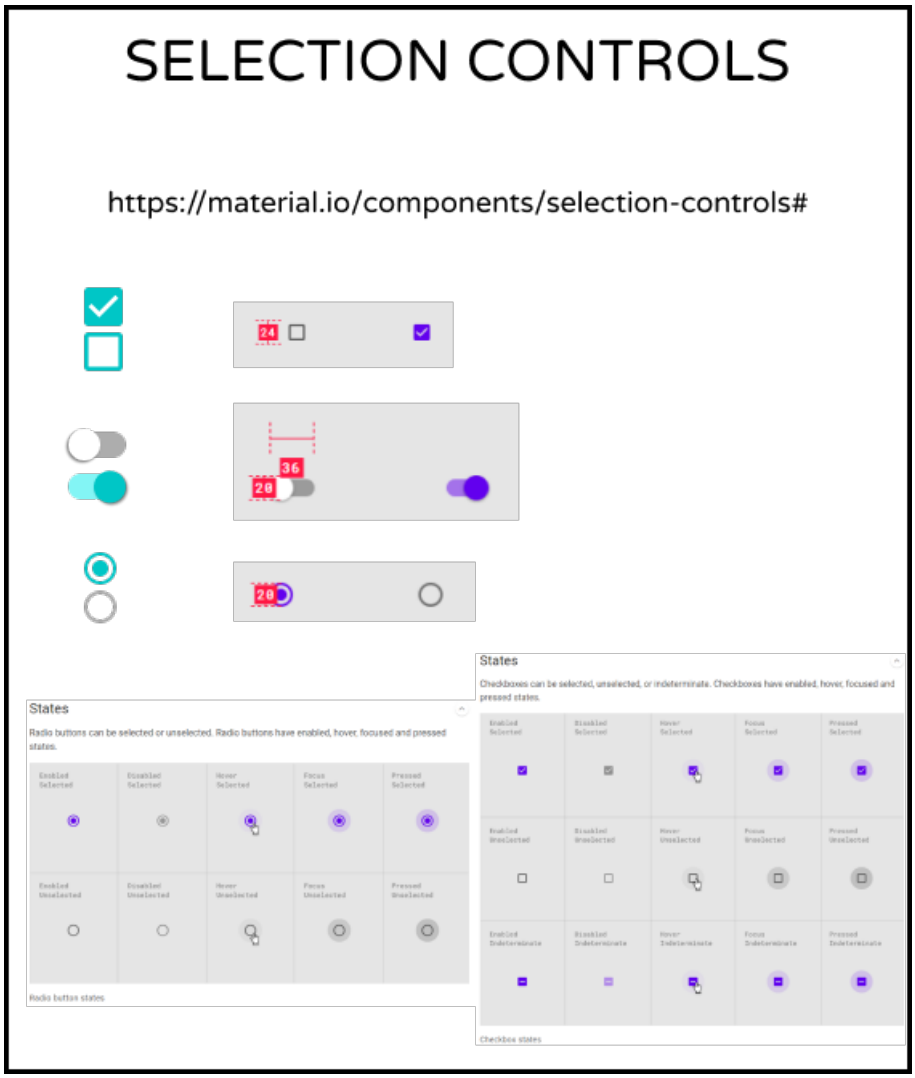


Figure C.11: Material Design based style convention for selection controls

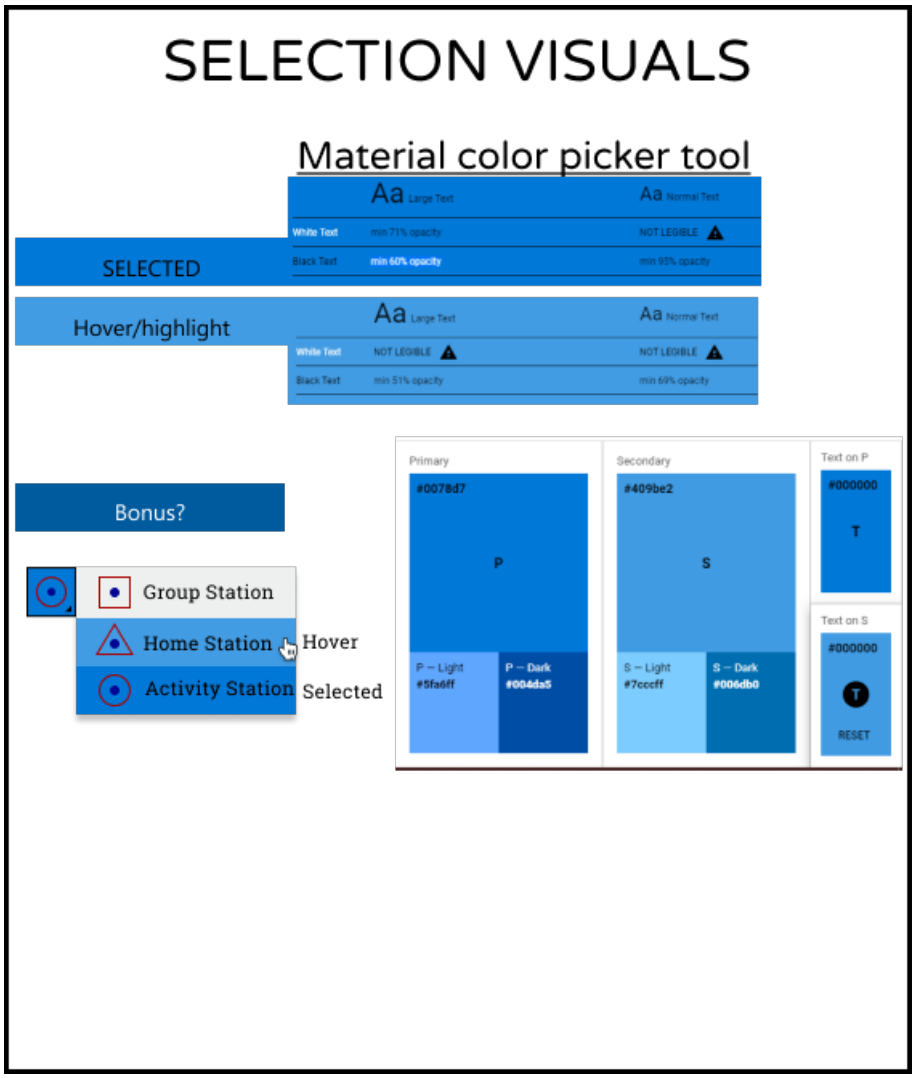


Figure C.12: Material Design based style convention for selection visuals

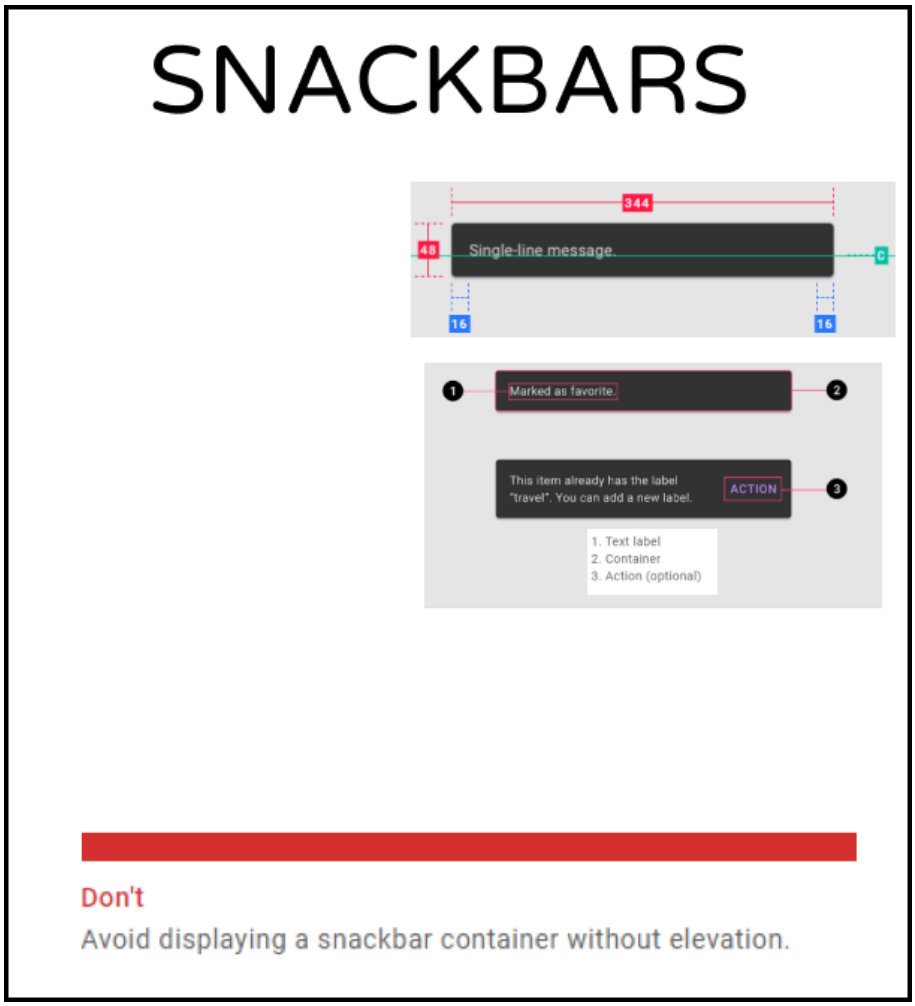


Figure C.13: Material Design based style convention for snackbars

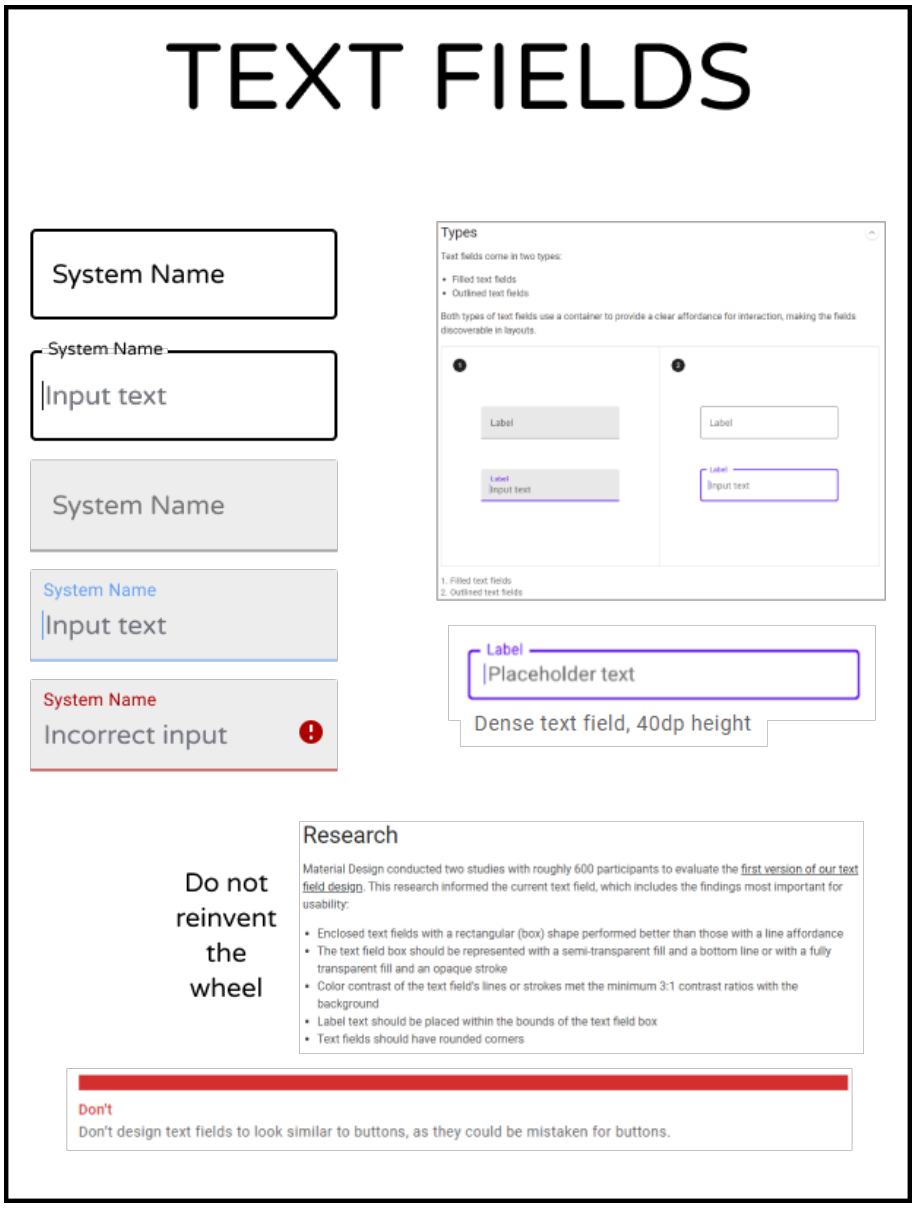


Figure C.14: Material Design based style convention for text fields

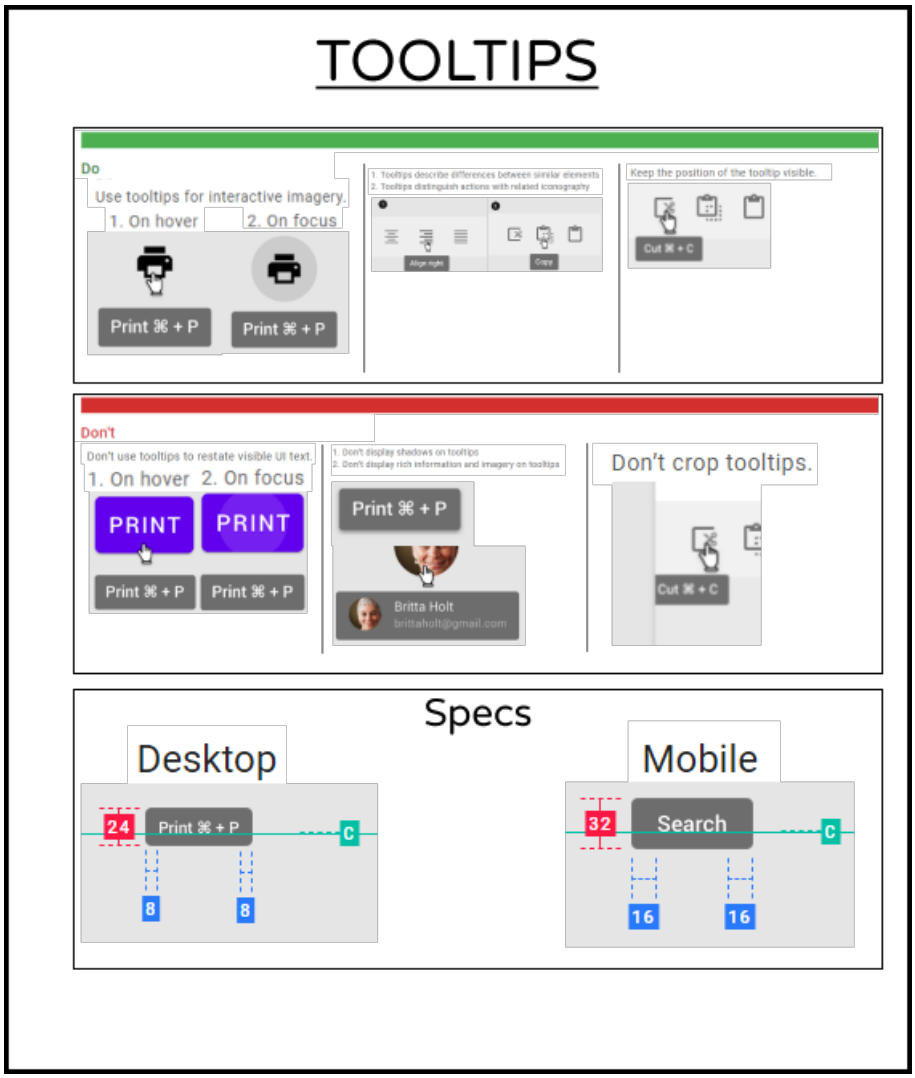


Figure C.15: Material Design based style convention for tool tips

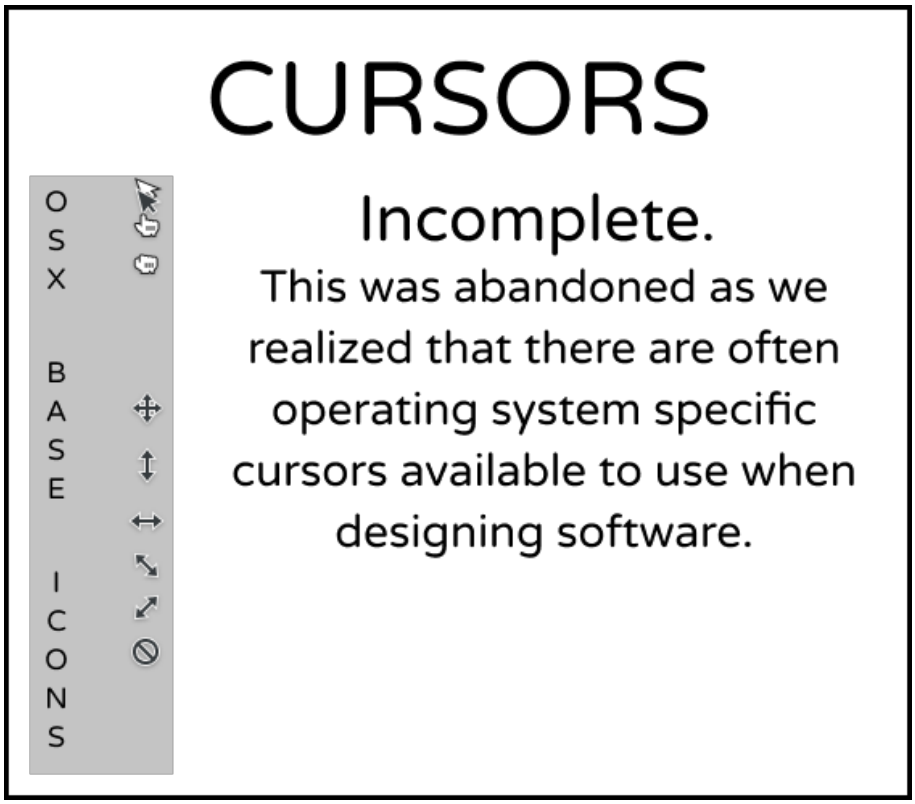


Figure C.16: Style convention topic we abandoned