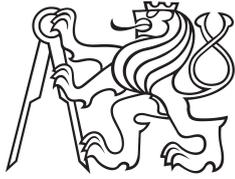


Master Thesis



Czech  
Technical  
University  
in Prague

**F3**

Faculty of Electrical Engineering  
Department of Computer Graphics and Interaction

## Production Process Visualization

**Bc. Anastasia Surikova**

Supervisor: doc. Ing. Zdeněk Míkovec, Ph.D.

Field of study: Open Informatics

Subfield: Human-Computer Interaction

January 2022



## I. Personal and study details

Student's name: **Surikova Anastasia** Personal ID number: **466389**  
Faculty / Institute: **Faculty of Electrical Engineering**  
Department / Institute: **Department of Computer Graphics and Interaction**  
Study program: **Open Informatics**  
Specialisation: **Human-Computer Interaction**

## II. Master's thesis details

Master's thesis title in English:

**Production process visualization**

Master's thesis title in Czech:

**Vizualizace výrobního procesu**

Guidelines:

People who work in the manufacturing industry have to deal with a large amount of stress daily. Late material delivery, violation of a product deadline, or inevitable human factor could be possible reasons behind this stress. This thesis aims to design a tool that supports the manufacturing process and eliminates workers' stress in the industry by using user-centered design techniques and production process visualization. The design process must consider an existing optimization algorithm for schedule creation and focuses only on designing the algorithm's output in a user-friendly way. Conduct and describe qualitative research with the target field representatives, followed up by creating the high-fidelity prototype based on research findings. Conduct usability tests of the prototype with the target user group and subsequently implement it with mocked backend.

Bibliography / sources:

1. Miovsky, Michal. (2006). Kvalitativní přístup a metody v psychologickém výzkumu. <https://www.martinus.cz/?ulem=24603>
2. Elizabeth Goodman, Mike Kuniavsky and Andrea Moed (2012). Observing the User Experience 3. Colin Ware. Visual thinking for design 4. Colin Ware. Information Visualization, perception for design 5. Bill Buxton . Sketching User Experiences 6. Miroslav Disman (2002). Jak se vyrábí sociologická znalost: Příručka pro uživatele.

Name and workplace of master's thesis supervisor:

**doc. Ing. Zdeněk Míkovec, Ph.D., Department of Computer Graphics and Interaction**

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **04.02.2021** Deadline for master's thesis submission: **04.01.2022**

Assignment valid until: **30.09.2022**

doc. Ing. Zdeněk Míkovec, Ph.D.  
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.  
Dean's signature

## III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

\_\_\_\_\_  
Date of assignment receipt

\_\_\_\_\_  
Student's signature



## Acknowledgements

I want to thank my supervisor, doc. Ing. Zdeněk Míkovec, Ph.D., for his valuable insights and guidance throughout this project. I also thank prof. Dr. Ing. Zdeněk Hanzálek for the original idea of this thesis and sharing his expertise on the subject. I thank Ing. Miloš Kodad, the principal of the František Křižík Higher Vocational School and Secondary Industrial School of Electrical Engineering, for his help in finding user research participants, and I thank all the participants as well. I thank my amazing friends, Valentina, Lukas, Martina, and Vojta, for their love and support. I thank my therapist for taking care of my mental health through the tough times, and I thank myself for making it through.

## Declaration

I hereby declare that I have written the submitted thesis myself, and I quoted all used sources of information.

The figures in this thesis use icons made by Flaticon[1]. Images of personas are made by Freepik[2].

Prague, 1. January 2022

## Abstract

People who work in the manufacturing industry have to deal with a large amount of stress daily. Late material delivery, violation of a product deadline, or inevitable human factor could be possible reasons behind this stress.

This thesis aims to design a tool that supports the manufacturing process and eliminates workers' stress in the industry using user-centered design techniques and production process visualization. The design process considers an existing optimization algorithm for schedule creation and focuses only on designing the algorithm's output in a user-friendly way. This thesis describes qualitative research with the target field representatives and the creation of the high-fidelity prototype based on research findings. Then, the conducted usability tests of the prototype with the target user group and the subsequent implementation with mocked backend are described.

**Keywords:** HCI, UCD, visualization, industry 4.0, production process, hybrid application, Ionic framework

**Supervisor:** doc. Ing. Zdeněk Míkovec, Ph.D.

Praha 2, Karlovo náměstí 13, E-321

## Abstrakt

Lidé, kteří pracují ve výrobním průmyslu, se musí denně potýkat s velkým množstvím stresu. Možnými příčinami tohoto stresu mohou být pozdní dodávky materiálu, nedodržení termínu zakázky nebo nevyhnutelný lidský faktor.

Tato práce si klade za cíl navrhnout nástroj, který podpoří výrobní proces a eliminuje stres pracovníků v průmyslu díky metodikám User-Centered Design a vizualizaci výrobního procesu. Proces návrhu musí předpokládat existující optimalizační algoritmus pro plánování rozvrhu a zaměřit se pouze na návrh výstupu algoritmu uživatelsky přívětivým způsobem. Práce popisuje kvalitativní výzkum se zástupci cílové uživatelské skupiny a vytvoření high-fidelity prototypu založeného na výsledcích výzkumu. Dále popisuje provedené testy použitelnosti prototypu s cílovou skupinou uživatelů a následnou implementaci s využitím simulovaného backendu.

**Klíčová slova:** HCI, UCD, vizualizace, industry 4.0, výrobní proces, hybridní aplikace, Ionic framework

**Překlad názvu:** Vizualizace výrobního procesu

# Contents

<b>1 Introduction</b>	<b>1</b>		
1.1 Motivation and goals	2		
1.2 User-Centered Design	3		
1.3 Used terminology	5		
<b>2 State Of The Art</b>	<b>7</b>		
2.1 Existing solutions	7		
2.1.1 MES PHARIS® by UNIS, a.s.	7		
2.1.2 WorkMonitor by Workmonitor s.r.o.	7		
2.1.3 Similarities between existing solutions	7		
2.2 Future solution mock-up	9		
<b>3 User Research</b>	<b>11</b>		
3.1 Methodology	11		
3.2 Population sampling	12		
3.3 Research findings	14		
3.4 Personas	16		
3.4.1 Primary persona - Adam	17		
3.4.2 Supplement persona - Barbora	19		
3.5 Problem scenarios	20		
3.5.1 Production delay	20		
3.5.2 Machine breakage	21		
3.5.3 Catching up on events	21		
3.6 Functional requirements	22		
3.7 Evaluation	23		
<b>4 Design Concept</b>	<b>25</b>		
4.1 Features distribution	25		
4.2 User scenarios	25		
4.2.1 Production delay	26		
4.2.2 Machine breakage	27		
4.2.3 Catching up on events	27		
4.3 User flow and system screens	28		
4.4 Machine's states definition	29		
4.5 Conclusion	31		
<b>5 User Interface Design</b>	<b>33</b>		
5.1 Machine's states design	33		
5.1.1 Machine is on	34		
5.1.2 Machine is on and delayed	35		
5.1.3 Machine is on and has an alarm	36		
5.1.4 Machine is on, delayed and has an alarm	37		
5.1.5 Machine is off	37		
5.1.6 Machine is off and delayed	38		
5.1.7 Machine is off and has an alarm	38		
5.1.8 Machine is off, delayed and has an alarm	39		
5.2 Production overview	39		
5.3 Order detail	41		
5.4 Daily events	42		
5.5 Problem management	42		
5.6 Notifications	43		
5.7 Schedule recalculation	43		
5.8 Design testing and discussion	44		
5.8.1 Conclusion	45		
<b>6 Implementation</b>	<b>47</b>		
6.1 Technology selection	48		
6.1.1 Mobile development	48		
6.1.2 Hybrid development	49		
6.1.3 Web development	50		
6.1.4 Selected technology	51		
6.2 Data model	52		
6.2.1 Schedule	52		
6.2.2 Machine	52		
6.2.3 Order	54		
6.2.4 Problem	54		
6.3 Production overview	55		
6.3.1 Machines' placement	56		
6.3.2 Switching between overview and detail	57		
6.4 Other features	58		
6.5 REST API specification	58		
6.5.1 Schedule endpoints	58		
6.5.2 Machine endpoints	59		
6.5.3 Order endpoints	60		
6.5.4 Problem endpoints	60		
6.5.5 Notification endpoints	61		
6.5.6 Daily events endpoints	62		
<b>7 Usability Testing</b>	<b>63</b>		
7.1 Participants	63		
7.1.1 Screener	64		
7.1.2 Number of participants	65		
7.2 Testing preparation	66		
7.3 Testing process	66		
7.4 Results	68		
7.4.1 Issues and recommendations	68		
7.4.2 Findings	70		
7.4.3 Satisfaction results	71		
7.4.4 Summary	72		

<b>8 Conclusion</b>	<b>73</b>
8.1 Future work.....	74
8.1.1 Implementing mocked parts of the system.....	74
8.1.2 Analyzing the impact.....	74
8.1.3 Extending the algorithm usage	74
8.1.4 Extending the system functionalities.....	75
<b>A User Research Interview Scenario</b>	<b>77</b>
<b>B Designed Screens</b>	<b>79</b>
<b>C Usability Test Tasks</b>	<b>85</b>
<b>D Post-Test Interview</b>	<b>87</b>
<b>E Usage Manual</b>	<b>89</b>
E.1 Prerequisites.....	89
E.2 Running the project.....	89
<b>F Bibliography</b>	<b>91</b>

## Figures

1.1 Waterfall life cycle (adopted from [3]). . . . .	3	5.9 Design of "off + delayed" machine state. . . . .	38
1.2 Iteration life cycle including user. . . . .	4	5.10 Design of "off + alarm" machine state. . . . .	39
1.3 Star life cycle (adapted from [4]). . . . .	5	5.11 Design of "off + alarm + delayed" machine state. . . . .	39
2.1 Screenshot of the MES PHARIS® system by UNIS, a.s.[5] . . . . .	8	5.12 Lo-Fi idea of the production overview. . . . .	40
2.2 Screenshot of the WorkMonitor system by Workmonitor s.r.o.[6] . . . . .	8	5.13 Hi-Fi prototype of the production overview with labeled parts. . . . .	41
2.3 Example of a Gantt-chart. . . . .	8	5.14 Problem state diagram. . . . .	42
2.4 Mock-up of the final solution. . . . .	10		
3.1 Distribution graph of participants' job positions. . . . .	13	6.1 System architecture with highlighted focus of the thesis. . . . .	47
3.4 Storyboard of production delay problem scenario . . . . .	21	6.2 Example of the production visualization hybrid application usage. . . . .	49
3.5 Storyboard of machine breakage problem scenario . . . . .	21	6.3 Calculation of the machine's placement in the production overview. . . . .	56
3.6 Storyboard of catching up on events problem scenario . . . . .	21	6.4 Approach for switching between overview and detailed machine view. . . . .	57
4.1 Storyboard of production delay user scenario . . . . .	26		
4.2 Storyboard of machine breakage user scenario . . . . .	27	7.1 Number of discovered usability problems according to the number of test participants. . . . .	65
4.3 Storyboard of catching up on events user scenario . . . . .	27	7.2 Usability testing setup. . . . .	68
4.4 User flow chart. . . . .	28		
4.5 Machine state timeline. . . . .	29	B.1 TV view of the production overview. . . . .	79
4.6 Schedule state timeline. . . . .	30	B.2 App view of the production overview. . . . .	80
4.7 Problem state timeline. . . . .	30	B.3 Detailed machine view after zooming in production overview. . . . .	80
4.8 Machine state timeline combined with the states of other entities. . . . .	30	B.4 Order detail view. . . . .	81
5.1 Scheme of machine UI element. . . . .	34	B.5 Daily events view. . . . .	81
5.2 Design of "on" machine state. . . . .	35	B.6 Problem list view. . . . .	82
5.3 CTA buttons available in zoomed-in machine view. . . . .	35	B.7 New problem view. . . . .	82
5.4 Design of "on + delayed" machine state. . . . .	36	B.8 Problem detail view. . . . .	83
5.5 Design of "on + alarm" machine state. . . . .	36	B.9 Notification list view. . . . .	83
5.6 List of problems available in zoomed-in machine view. . . . .	37	B.10 Schedule recalculation dialog - step 1. . . . .	84
5.7 Design of "on + alarm + delayed" machine state. . . . .	37	B.11 Schedule recalculation dialog - step 2. . . . .	84
5.8 Design of "off" machine state. . . . .	38		

## Tables

3.1 Distribution of job positions among the research participants. . . . .	14
4.1 Distribution of system features among two views. . . . .	26
4.2 Flow of events connected to the changes in machine state. . . . .	31
5.1 Visible features in overview and detail of machine state . . . . .	34
7.1 Usability testing screener . . . . .	64



# Chapter 1

## Introduction

Industry 4.0 is a concept introduced during the industrial trade fair Hannover Messe in 2011[7]. It defines the “Fourth Industrial Revolution” as a fusion of technologies blurring the lines between the physical, digital, and biological spheres. Industry 4.0 is characterized by integrating new technologies, such as Big Data, cloud computing, artificial intelligence, or the Internet of Things in the manufacturing field. The main idea behind Industry 4.0 is to develop a smart factory, which operates as a cyber-physical production system using new technologies to complete day-to-day tasks.

Industry 4.0 solutions can be applied in various stages of the manufacturing process, including planning and scheduling. Finding an optimal production schedule is crucial for minimizing expenses and increasing the final product’s quality. It is also one of the most common combinatorial problems in computer science. One can choose between various scheduling algorithms and calculate an optimal sequence of tasks based on the given parameters. Using such an algorithm in production will automate the planning process, help cope with unexpected situations, and bring the production closer to a smart factory.

According to the case study conducted in 2019, some Czech companies and multinationals with manufacturing plants in the Czech Republic have set ambitious goals for automation, digitization, and new digital services, but at present most activity is restricted to pilot scale with no wide-scale reconfiguration or transformation at the company level[8]. In many cases, the complexity of offered I4.0 solutions or the need to hire highly skilled workers to integrate such solutions discourages companies from using them.

The aim of this thesis is to develop a visualization of an output of a scheduling algorithm for manufacturing businesses. The solution will attempt to create a new way of displaying the calculated schedule, so the visualization is easy-to-read and will require no additional training before its usage.

This thesis consists of the following chapters:

- Chapter 2 describes the state of the art analysis and introduces the idea of the final solution.
- Chapter 3 describes conducted user research and defines the list of the functional requirements for the final solution.

- Chapter 4 discusses the general approach for the visualization tool design and how it will contribute to the end-user.
- Chapter 5 goes over the design process of each feature of the final solution and describes the design testing process.
- Chapter 6 describes the implementation process of the high-fidelity prototype and defines requirements for the API.
- Chapter 7 is focused on the usability testing of the implemented prototype and its results.
- Chapter 8 concludes the thesis and discusses future work.

## 1.1 Motivation and goals

According to the case study[8], despite ambitious goals to automate manufacturing processes, many Czech companies struggle to integrate innovation. The goal of this thesis is to support manufacturing companies in adopting the I4.0 principles, in particular, encourage them to use scheduling algorithms in the planning phase of the manufacturing process by offering them an intuitive and easy-to-use visualization of the calculated production schedule. The final solution will enrich existing operations and ease the company's transition into I4.0 format.

Creating a solution that enables the setting up of a scheduling algorithm and shows an interactive visualization of the calculated schedule is quite a big task. Since settings for an algorithm can vary for different companies and types of productions, this thesis will not be focusing on this part of the tool; nor will an algorithm be described in this paper. Algorithms used to determine a schedule can differ, but the results they give are the same. The calculated output of an algorithm consists of tasks' assignment to the sources, time slots for each task, and additional information about possible problems of the schedule. The goal is to go beyond basic techniques for displaying a schedule and offer the viewer a comprehensible and accurate visualization of all the aspects of an algorithm's output.

Besides the visualization part of the tool, this thesis aims to determine in which ways this type of solution can contribute to daily operations and support workers in overcoming stressful situations that occur during their workday in the manufacturing field.

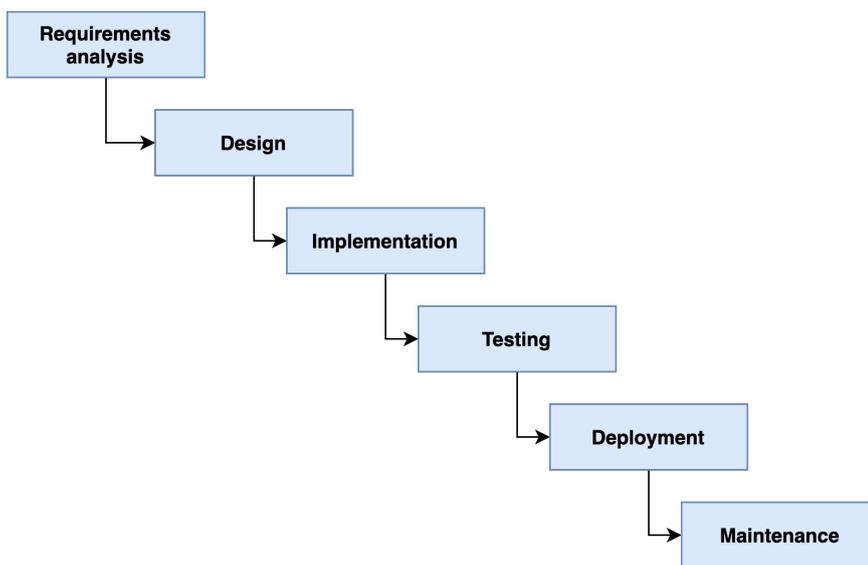
The overall list of this thesis's goals is:

- Introduce a novel idea of displaying the calculated production schedule.
- Determine the functional requirements for the final visualization tool.
- Design and test the prototype of the visualization tool.
- Implement the front-end part of the prototype, including the adjustments determined by the testing.

- Conduct usability testing of the implemented front-end.
- Conclude the progress and set the goals for the future development.

## 1.2 User-Centered Design

Traditionally software development has had a cascade life cycle in which each phase represents an independent part. Once the phase is ended, the development continues to the next phase until the process is completed. This life cycle is often called waterfall and is presented in Figure 1.1.



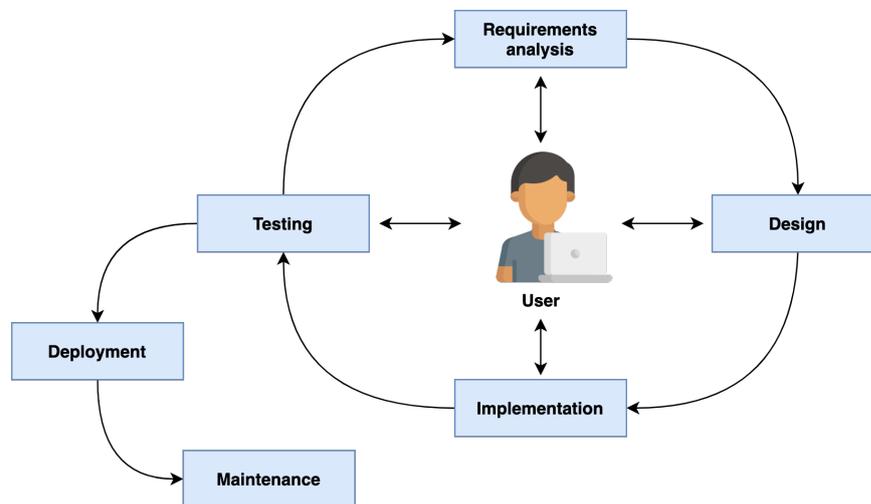
**Figure 1.1:** Waterfall life cycle (adopted from [3]).

This approach, however, brought such issues as an inability to change requirements once the development has started or a difficulty to indicate the source of mistakes when the product is delivered. Waterfall proposes to test the system at the very end after the system is developed, which leads to discovering errors that could have been prevented earlier. The imperfections of the waterfall model have caused the creation of an interactive life cycle. In practice, these stages overlap and feed information to each other. During design, problems with requirements are identified. During coding, design problems are found, and so on. The software process is not a simple linear model but involves feedback from one phase to another [9]. The user-centered design uses the iteration life cycle and includes the end-user in each stage of the process, ensuring every user requirement is met (see Figure 1.2).

To create a visualization tool that will not interrupt existing production processes and can be integrated with no need of hiring high-skilled workers,

the User-Centered Design techniques must be applied. These techniques put the user in the center of the design or development process, making his or her emotions, needs, and mental processes the main focus. This requires gathering knowledge about the user and the context in which the final product will be used. This stage of the development process will be discussed later in Chapter 3 of this thesis.

Acquired knowledge about the user is used to define user requirements, creating the prototype's foundation. The prototype is tested in terms of whether it serves defined user needs, and only after this test can the design be implemented. The same test is applied continuously during the implementation process to determine whether the emerging product fulfills its purpose.



**Figure 1.2:** Iteration life cycle including user.

An adapted star life cycle was proposed in 1993[4], in which the central point is the **evaluation**, due to its relevance in every stage of the development (see Figure 1.3). Evaluation is conducted primarily against the needs of the user, that is why acquiring data about the user is crucial at early stages. Additionally, authors describe that life cycle is "intended to be equally supportive of both top-down and bottom-up development, plus inside-out and outside-in development", therefore the process can start at any phase.

This thesis will apply one of the star life cycle principles and start with the **prototyping** phase by developing the first mock-up of the visualization tool. It will come in handy during the user research to explain to the participants the main aim of the final tool. It will be clearer for them what part of the manufacturing processes they should focus on during the user research interviews.

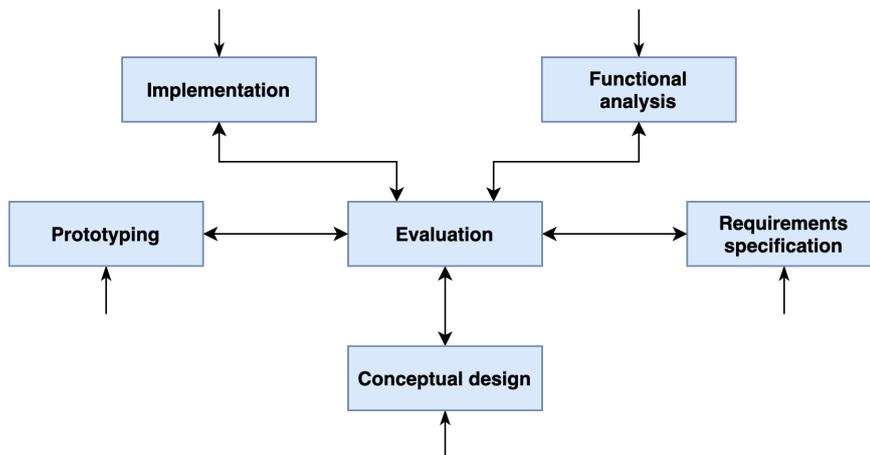


Figure 1.3: Star life cycle (adapted from [4]).

## 1.3 Used terminology

**The user.** As said before, User-Centered Design involves the user, or the end-user, in each stage of its process. Although, the term “end-user” could be interpreted in different ways, such as:

- a customer who would buy the final product,
- a person who will use the final product but not necessarily buy it,
- other people or institutes who have an interest in developing the product (often described as “stakeholders”).

In this thesis, the term “user” will describe people who will use the system to perform tasks and achieve goals. In the context of the manufacturing field, in which the visualization tool will be integrated, the term “user” will refer to the workers of the manufacturing company.

**Mock-up.** Mock-up is a static design that represents the idea of the final visualization tool[10]. It does not necessarily show all the required features, instead, it outlines the concept of what the final solution may look like. In this thesis, a mock-up’s creation and usage is described in Chapter 2.

**High fidelity (Hi-Fi) prototype.** High fidelity prototype (opposed to low fidelity, or Lo-Fi) is more advanced, with its aesthetics and functions far closer to that of the final product[11]. Hi-Fi prototypes could be represented as a clickable design or partly programmed application. This thesis uses both types. By the end of the design process, a clickable design is evaluated with the users (see Chapter 5). As a result of this thesis, the programmed application with the mocked data is created.



## Chapter 2

### State Of The Art

This chapter describes the analysis of the existing solutions used in the manufacturing businesses for the production process visualization. Then the idea of the future solution is described and presented in the mock-up.

#### 2.1 Existing solutions

One of the goals of this thesis is to develop a visualization that offers a novel way of displaying the calculated production schedule. However, a good starting point would be to analyze existing solutions that offer similar functionality of displaying the state of production or planned schedule.

##### 2.1.1 MES PHARIS® by UNIS, a.s.

MES PHARIS® is a full-fledged production system for the preparation and management of production. As seen in Figure 2.1 the data are displayed in a table-like way, with the time slots on the x-axis and the machines on the y-axis. The main content of the tool displays the processes assigned to the machines at the given time slot. A detailed view of each process is available, either upon mouse click or mouse hover.

##### 2.1.2 WorkMonitor by Workmonitor s.r.o.

The WorkMonitor system is designed to display the status of production machines, monitor their operation, and calculate their work cycles. From the layout point of view, the system is similar to the MES PHARIS® system by UNIS, a.s. The x-axis is dedicated to the selected time period, whereas the y-axis displays the list of the machines. On the right side of the system, the general information for each machine is displayed. This system has control buttons to manipulate the view (see Figure 2.2).

##### 2.1.3 Similarities between existing solutions

It is clear to the viewer that all existing solutions resemble a Gantt-chart layout in their user interface. Gantt-chart is a type of bar chart that is mainly

## 2. State Of The Art

used to visualize schedules and show dependencies between events. This type of chart was designed by Henry Gantt in the beginning of the 20th century[12]. The example of a Gantt-chart is shown in Figure 2.3.

Similar to a Gantt-chart, existing systems use the x-axis for the time and the y-axis for the machines list. The main part of the layout displays the assignment of a process to the machine in the given time slot.

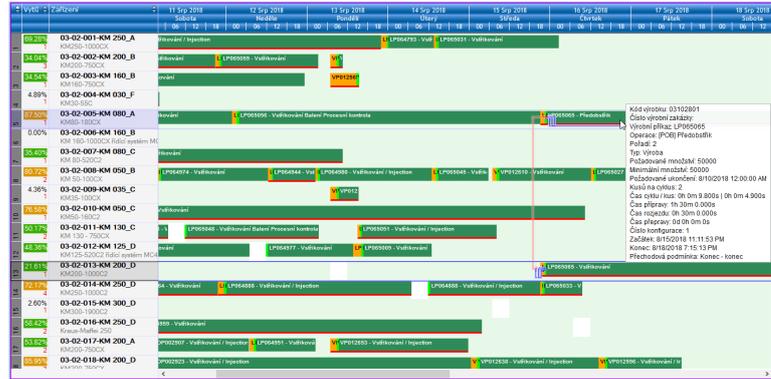


Figure 2.1: Screenshot of the MES PHARIS® system by UNIS, a.s.[5]

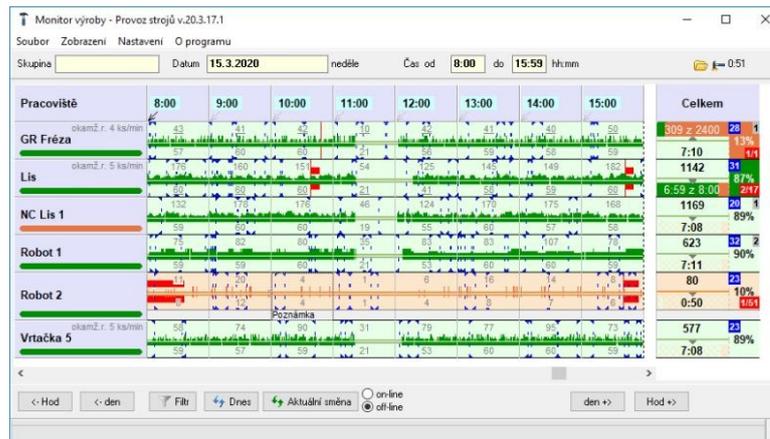


Figure 2.2: Screenshot of the WorkMonitor system by Workmonitor s.r.o.[6]

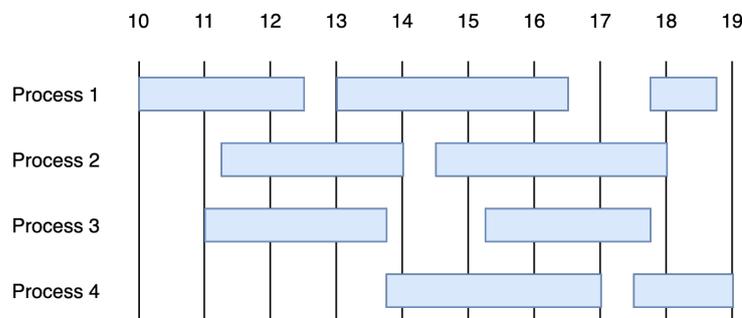


Figure 2.3: Example of a Gantt-chart.

## ■ 2.2 Future solution mock-up

This thesis aims to design a visualization of the calculated production schedule that is easy to understand and does not require any additional training to use it. Systems that apply Gantt-chart layout may be challenging to read for the users who did not work with these types of data presentations before. Since one of this thesis's goals is to encourage manufacturing companies to use optimization algorithms, the design of such algorithm's output must be easy-to-read even for the people who are not proficient with data visualization or computer science in general.

The idea behind the final visualization tool is to project the algorithm output onto something known for every worker at the production - **layout of the production hall**. The current production state will be visualized as the states of individual machines placed according to their real position in the factory (see Figure 2.4). This type of layout also allows the display of such additional information as the state of the warehouse.

Time is an essential variable for displaying time-varying data, such as calculated production schedules. For this purpose, the mock-up introduces the **interactive timeline**. The user can use the timeline to explore the calculated schedule (move forward in time) or see the history of events that occurred in the past (move back in time). As the user changes the position of the timeline pin, the states of the machines will change accordingly to the events in the focused moment.

**The right side** of the mock-up leaves an open space for the additional design elements. There could be a filter for the machines, a filter of the orders that are being currently executed, or a detailed overview of the displayed state. Since it is only a mock-up of the final solution, leaving such open questions is a good practice. The goal of the user research described in chapter 3 is finding the answers to these open questions.

Another goal of the mock-up is to help the user research participants understand what part of the production process they should focus on. Some participants may also be motivated by the printed-out artifact to think of other features they would find helpful in the system. This information will give the researcher valuable insights into the ways people in the manufacturing industry preserve the production process and think in general.

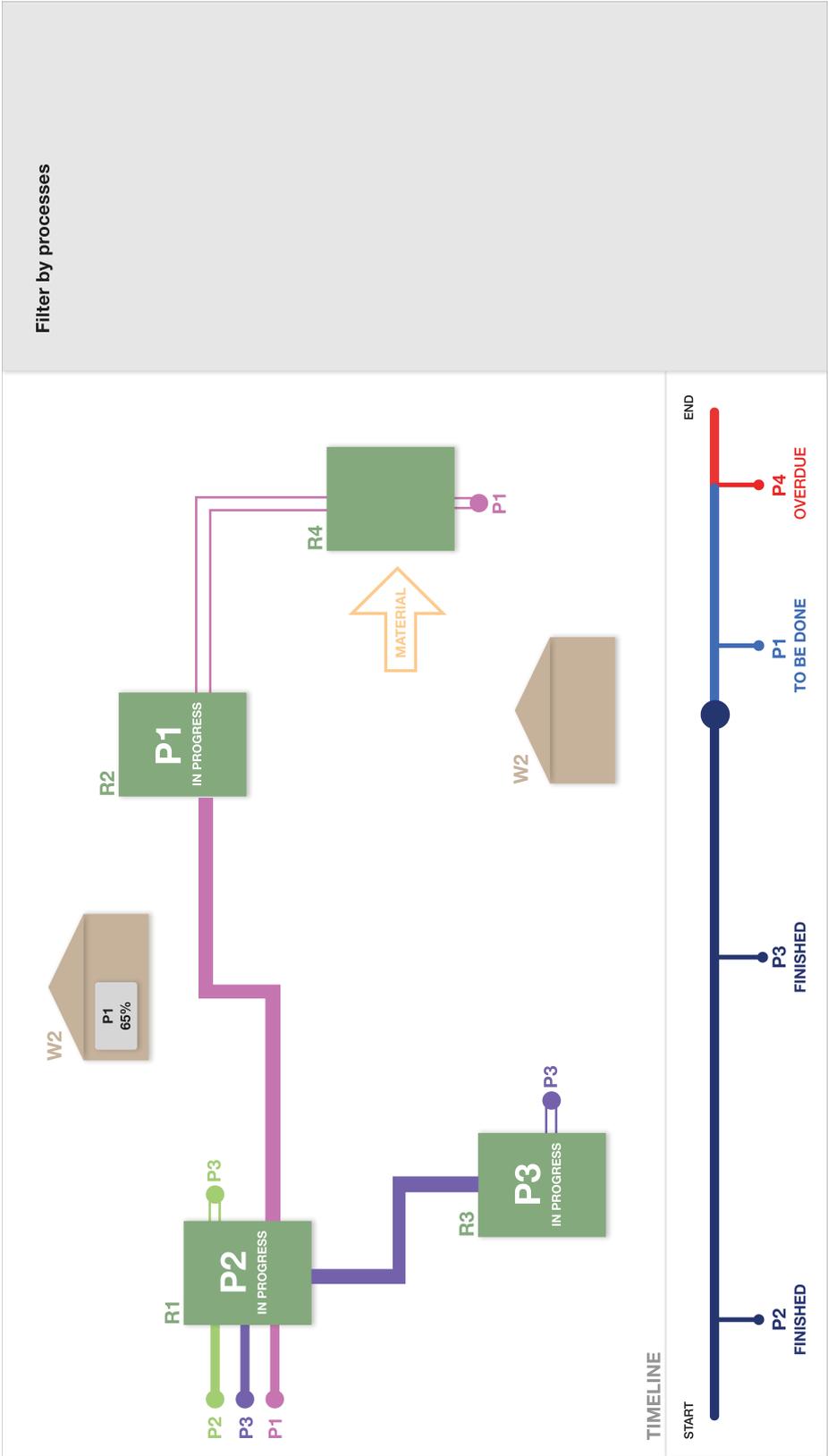


Figure 2.4: Mock-up of the final solution.

# Chapter 3

## User Research

When applying User-Centered Design techniques, one must understand the user's emotions and feelings. Gaining empathy for the user and their needs could be essential in software development, especially when it must serve a specific industry, such as manufacturing. User research helps uncover valuable insights about both the target group of users and the context of the given problem. In order to create a visualization of production that will enrich existing processes in the manufacturing businesses, five research questions must be answered:

1. What is the workers' mental model of the manufacturing process?
2. What is the people's role in the manufacturing process?
3. What is the software's role in the manufacturing process?
4. Which problems are encountered during the manufacturing process?
5. How are these problems solved?

The answers to these questions give the researcher an insight into the current state of manufacturing processes as well as people's role in it. Understanding the mental model of the manufacturing process (question 1) will help design a visualization aligned with the existing idea of production flow so that no additional training is required for workers. Discovering the variety of solvable problems (question 4) and how they are addressed now (question 5) will suggest the functional requirements for the final system. Exploring the size of a person's impact on production (question 2) will reveal the number of problems that could not be solved by software and optimization. Discovering the current usage of software solutions in production processes (question 3) will help determine the companies' readiness for I4.0 integration and suggest how the final visualization tool could make this integration easier.

### 3.1 Methodology

There are multiple techniques to obtain knowledge about the users and the environment in which the system will be used. Those techniques include:

- observing users,
- interviewing users,
- or gaining knowledge from users through questionnaires.

Data collection through questionnaires is classified as a quantitative method, whereas observing or interviewing users as qualitative methods of user research. Selecting the correct approach or design depends on the question posed or the phenomenon under exploration[13]. Since the main goal of user research conducted in this thesis was to understand the people who work in the manufacturing field, find out what kinds of problems they experience, and gather requirements for the visualization tool, the qualitative method must be chosen.

The data were collected by semi-structured interviews, sometimes combined with the techniques of the observations. When it was possible to meet for the interview on the manufacturing site, the researcher observed potential end-users at their work, which gave a deeper understanding of what a workday in the manufacturing field looked like. Some of the conducted interviews were held remotely by video call. Each interview lasted 40 to 60 minutes. The interviews followed the interview scenario presented in Appendix A. Every question in the scenario contributed to the five main research questions described in Chapter 3.

The order and wording of the questions were changed to fit the specific company and the interviewee's role in it. Additional questions were added depending on the information the interviewee gave[14]. At the same time, the nondirected interviewing principles were applied so that no answer was biased by the researcher[15]. The interviewee's thoughts and emotions were obtained without being filtered through the researcher's preconceptions.

Before the start, the researcher acquainted the interviewees with the research goal and asked permission to take notes during the interview. The researcher presented the interviewees with the mock-up presented in Figure 2.4 to specify the primary focus of the interview. After the interview, the interviewees were asked if they were interested in participating in the usability testing in the future. If so, the researcher contacted them during the testing phase of this work.

## 3.2 Population sampling

Finding the right participants is crucial for successful user research. The first step is to define the population representing potential users of the system. The next step is to draw a *sample* - a few individuals representing that population during the research.

Since this work focuses on creating a visualization tool of the manufacturing process, people that work in manufacturing companies make up the research population. Representatives from specific job positions create a population sample — the research targets such positions as business owners, production

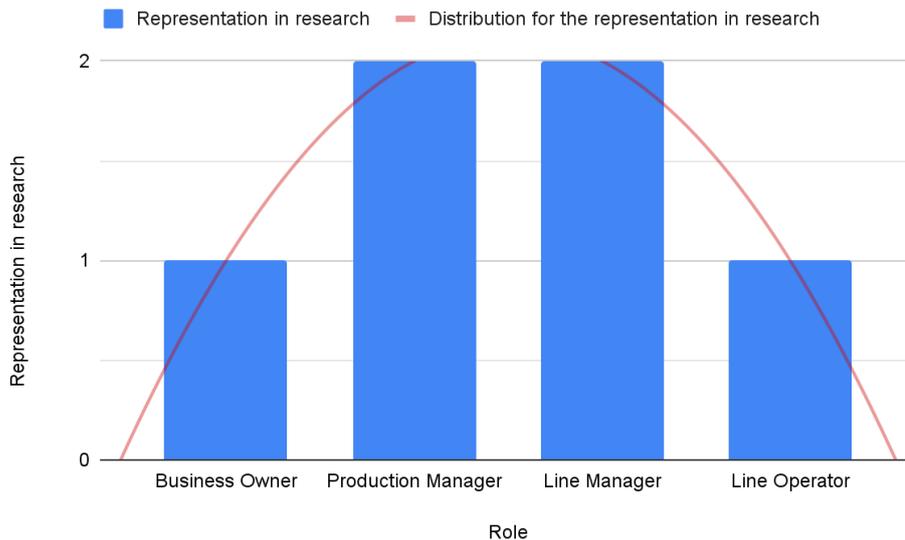
managers, line managers, and line operators. At least one representative from each position is enough to cover the whole range of possible problems encountered during the manufacturing process.

Ideally, a researcher should use probability sampling methods to sample the population[16]. These methods reduce the risk of sampling bias, which occurs when the probability of being selected is higher for the specific individuals. The systematic probability sampling was applied by contacting every second manufacturing company listed in the public catalog of verified companies in the Czech Republic[17]. This approach has an exploratory nature that helps uncover unexpected aspects of the investigated field, such as new use-cases for the final visualization tool.

The researcher also used the non-probability method - the snowball - to sample the population. Each research participant was asked to nominate another subject known to them to participate in the research. This approach made the population sample increase like a rolling snowball, which corresponds with the method's name.

A total of six participants took part in the research. Since the qualitative approach was applied, the number of research participants did not have to be large. The knowledge collected from participants was more important than their number. The distribution of job positions among the research participants is described in Table 3.1.

Figure 3.1 clarifies that the research focused more on the roles of production manager and line manager. These roles have the biggest probability of taking part in the production planning process, and therefore, are more likely to represent the end-user of the final visualization tool.



**Figure 3.1:** Distribution graph of participants' job positions.

Job position	People count
<p><b>Business Owner</b></p>  <ul style="list-style-type: none"> <li>■ owns a production business</li> <li>■ takes production orders from clients</li> </ul>	1
<p><b>Production Manager</b></p>  <ul style="list-style-type: none"> <li>■ takes care of the whole production</li> <li>■ gives tasks to line managers depending on currently executed order</li> </ul>	2
<p><b>Line Manager</b></p>  <ul style="list-style-type: none"> <li>■ takes care of the production of a single product on one production line</li> <li>■ gives tasks to line operators</li> </ul>	2
<p><b>Line Operator</b></p>  <ul style="list-style-type: none"> <li>■ a blue-collar worker</li> <li>■ operates machines</li> <li>■ does tasks assigned by the line manager</li> </ul>	1

**Table 3.1:** Distribution of job positions among the research participants.

### 3.3 Research findings

The research aimed to answer the five main research questions and bring a deeper understanding of current production processes and the workers' problems in manufacturing companies. Although companies, that took part in the research, manufactured different types of products and were of different structures and sizes, the key research findings can be generalized for all of

those companies. These findings are represented in the list below, which will serve as a source of information for future analysis and development of the visualization tool.

- A production process is tied to a customer's **order**, consisting of the amount of product to be manufactured and the **deadline**.
- Production is planned and scheduled by hand in all interviewed companies; no optimization algorithm is used. The most advanced tool that is used for production planning is Microsoft Excel. In some companies, a planning department produces a **production plan** (but uses no algorithms either).
- Unlike production planning, the **current production state** is usually supported by software solutions, such as SAP and MES. Several companies that participated in the research had a TV screen in the production hall, where the current state of machines was displayed. The same information was also accessible by the company's internal application. The current state of machines is usually displayed in a table.
- A manager responsible for **human capacities** plans shifts (by hand), so the customer's order is fulfilled and the deadline is met. Workers are often trained to work with a specific machine. This fact is considered during the shifts planning.
- The workers often feel overworked and have no vote in the **shift's planning**.
- One of the main struggles of production managers is **quality control** of the final product. Some of the interviewed participants suggested that checking the **material** before it goes into production would be a great precaution.
- Managers would like to be **notified** about the increased scrap rate of the machine. That way, the service assistance could be called before severe **damage** occurs.
- People play a huge role in production problems. Late material delivery, line operator running late to work - all of that affects and **delays** the production. The only current solution is to work more instead of changing the production plan (which is understandable since changing the plan is done by hand).
- Managers can not let go of their phones since they must act upon every **problem** in production. Some managers said that most of these problems could be solved by workers themselves, but instead, they would rather contact their superior.

## 3.4 Personas

The research findings are extensive and come from various participants: from company owners to blue-collar workers, representing several segments of the target user group. Since every interviewee has a different placement in the company hierarchy, and every company has a different size and manufacturing methods, keeping in mind all the aspects when designing a solution is quite difficult. In this situation, personas come to help.

Persona is a collective picture of the end-user - his archetype. It unifies all the knowledge acquired during the research and helps designers emphasize with the final user[18]. Persona can be perceived as the user brought to life. The more detailed description a persona has, the more helpful it is for making decisions about functionalities, interaction, and visual elements in the design.

Sometimes, however, the end-user is not the only one affected by the system which is being developed. Integrating a new software into a manufacturing business will most certainly affect the company's processes, which involve many more other people besides the end-users of the software. There are several types of personas[19]:

- **Primary persona** - to represent the end-user who will use the system to achieve goals and complete tasks.
- **Secondary persona** - to represent a special group of end-users who might want some additional special features of the software (i.e., users with visual impairments).
- **Supplemental persona** - to represent a group of people whose needs are aligned with the ones of the primary and secondary personas and are completely satisfied by the system designed for the end-users. Supplemental persona completes the comprehensive representation but is not the focus during design.
- **Customer persona** - to represent the product's customer who is not the end-user.
- **Served persona** - to represent a group of people who will not directly use the system but will be affected by its work.
- **Negative persona** - to represent a group of people who are not the target group for the design.

In the user research, the roles of production manager and line manager were represented more than the others. Both roles take part in the production planning process, have similar goals, and in some businesses may be fused into a single job position. A production manager distributes customers' orders among the line managers, who then plan the production for the machines in the production hall. For the final visualization tool, the role of line manager would be represented by the primary persona, and the role of production

manager by the supplemental persona, who might sometimes use the final tool, but will not be the focus of the design.

### ■ 3.4.1 Primary persona - Adam

The primary persona - Adam - represents the potential user, who has an overview of the production, manages a team of workers, and reports to the production manager Barbora represented by the supplemental persona in Subsection 3.4.2. Adam's background, goals, frustrations, and motivations are presented in the card below, along with his current set of software skills.



#### **Name**

Adam Svoboda

#### **Age**

50

#### **Family status**

married, 2 children (20 and 18 years old)

#### **Hobbies**

Fishing, bike riding

#### **Occupation**

line manager at Soaps&Washes company

---

### **Background**

Adam is a line manager in a Soaps&Washes company. He manages 1 line of dish soap production. Several machines are placed along the line: some pour the dish soap into containers, some put the lid on, some handle packaging. Some machines have their operators, who change their settings according to the currently executed order. Adam monitors the state of machines and the readiness of the currently executed order in his office next to the production hall. Besides machine operators, Adam also manages workers at the end of the line, whose responsibility is to perform quality control of the final product. Adam ensures the product is successfully loaded into the trucks and transported to the correct address. After the order is finished, he plans a production schedule for the next order assigned to him by the production manager Barbora. Adam reports the production state to Barbora every week.

---

### **Goals**

- to plan production schedule so the order is fulfilled and the, deadline is met,

- to have enough workers to operate the machines,
- to make sure every worker has a work to do,
- to have an overview of the current production state,
- to be able to track which machine has caused the damage to the final product,
- to be alerted when the machine has an increased scrap rate,
- to be alerted when it is time for the planned machine service,
- to monitor the active-time / inactive-time ratio of every machine,
- to deliver a good report for the top management,
- to know how much material needs to be ordered and when,
- to store only as much material as needed for the current plan,
- to know when the product is almost ready to check on its quality before shipping.

---

### **Frustrations**

- having to seek the cause of production delay,
- new employees damaging the machines because of the lack of training,
- having to tell workers what to do that day,
- not knowing where the truck with the material is when it is late,
- discovering bad material quality in the middle of production,
- having to deal with the problems that could be solved by the workers,
- having to plan the schedule by hand in Excel,
- not knowing what happened when he returns to work from time off,
- having to contact MES support in case of every change in the production process.

---

### **Motivations**

- fewer responsibilities,
- not needing to learn new technology,
- awareness of the production state,
- awareness of events and problems,

- better communication inside the team.

---

### Skills

- Microsoft Office,
- MES (Manufacturing Execution System),
- SAP (Enterprise Resource Planning system — ERP),
- Internal information system running on the web platform.

### ■ 3.4.2 Supplement persona - Barbora

The Barbora persona complements the Adam persona by giving more context to the production processes in the sample manufacturing company. The key characteristics of the supplementary persona are always related to the primary persona. Highlighting the differences and similarities between two personas supplies the design process with a more comprehensive end-user representation. The Barbora persona will not, however, be the focus during design.



#### Name

Barbora Nováková

#### Age

37

#### Occupation

production manager at Soaps&Washes company

---

### Background

Barbora is a production manager in a Soaps&Washes company. She manages five lines of production. Each line produces a different kind of wash product. One of Barbora's subordinates is line manager Adam, who handles the production of dish soap. Barbora assigns some of the company's orders to Adams and gets a report on the state of production every week. Every month, Barbora reports the production state to the top management of Soaps&Washes company.

---

### Unlike Adam, Barbora...

- ...do not plan a production of a single order.
- ...is not aware of the daily schedule on the production line.

- ... does not communicate with line operators.
- ... does not plan shifts.
- ... does not handle machine breakage with technicians.

---

**Like Adam, Barbora...**

- ... cares about delivering all orders in time.
- ... reports the state of production to her management.
- ... wants to know the state of production on demand.
- ... deals with the material delivery and state of the warehouse.
- ... deals with order's transportation.
- ... is frustrated by having to deal with the problems that could be solved without her.

## 3.5 Problem scenarios

Adam's persona answers the question of **who** will use the final visualization tool. The question of **when** the tool will be used requires the creation of the problems' scenarios[20]. Scenarios are typical for an ideation stage of the design thinking process. Since this stage uses user-centered techniques, problem scenarios will be focused on the end-user — in the case of this thesis — production manager Adam. The goal is to map some of Adam's goals, motivations, and frustrations to specific situations where the visualization tool would come in handy.

Storyboards help to visualize the problem which is described in a scenario[21]. They consist of several boxes with illustrations, which resemble comics. Characters do things and illustrate emotions according to the scenario, to help designers understand the end-user better. Every problem scenario is supported by its storyboard<sup>1</sup>.

### 3.5.1 Production delay

Adam walks through the production hall when he notices some line operators panicking around one of the machines. "The truck with the material is late" - an operator explains to Adam - "We cannot start the production". Adam calls the delivery company and discovers that the material will be delivered in about 2 hours. Adam asks the operator to stay at work longer that day to fulfill the production plan. The operator is disappointed because he will miss the family dinner, and Adam is stressed because there is no other way to handle the delay and meet the order's deadline.

---

<sup>1</sup>Storyboards were created on StoryboardThat.com[22]



Figure 3.4: Storyboard of production delay problem scenario

### 3.5.2 Machine breakage

Adam sits in his office when his phone rings. “The cutting machine is broken,” says a line operator on the line — “I do not know what to do.” Adam has to call a technician at the factory to tell him to look at the broken machine. After a while, the technician calls back to say that the machine is working again. Although the problem is solved, Adam is frustrated that he lost his time handling the problem, which could have been handled without him.



Figure 3.5: Storyboard of machine breakage problem scenario

### 3.5.3 Catching up on events

Adam walks into the production hall after his week-long vacation. He has a full day of meetings in front of him to catch up on events at the production, the state of the orders, and the plan for the next week. Adam feels exhausted already, knowing his productivity will be close to zero today.



Figure 3.6: Storyboard of catching up on events problem scenario

## 3.6 Functional requirements

The analysis of research findings and problem scenarios has resulted in a list of the functional requirements for the system. According to the gathered information from the conducted user research, it is requested for the final visualization tool to offer these functionalities:

- **Production management**
  - **Past events**
    - Displaying past events in the production
  - **Current state**
    - Displaying the current state in a easy-to-understand way
    - Splitting the information into overview and detail
  - **Future schedule**
    - Setting the deadline and priority of orders
    - Calculating the optimal schedule
    - Displaying a daily plan
- **Problem management**
  - Notifying when the problem occurs
  - Assigning responsibility to resolve a problem to the workers
  - Recalculating new schedule when necessary
- **Machine management**
  - Handling breakage
  - Displaying the machine's state
  - Displaying increased scrap rate
  - Displaying information about service assistance
- **Human resources management**
  - Shifts planning
- **Material management**
  - Displaying the warehouse state
  - Notifying when there is a material shortage
  - Notifying to check on material quality when it arrives
- **System settings**
  - Setting up the production layout
  - Setting up the algorithm's conditions

Regardless of the small number of research participants, the research brought up a wide range of problems and ideas. Due to the time shortage, only some of the parts will be designed and implemented in this thesis as listed below:

- **Production management**
- **Problem management**
- **Machine management**
- Human resources management - will not be implemented in this thesis
- Material management - will not be implemented in this thesis
- System settings - will not be implemented in this thesis

However, these parts are also the topic of interest of the users that must not be neglected in future work on the product of this thesis.

## ■ 3.7 Evaluation

Created problem scenarios and functional requirements were evaluated with the research participants, whose company's production processes and organization structure fitted into the described case. The evaluation did not bring out any necessary changes. The development process could proceed into the design stage.



## Chapter 4

### Design Concept

This chapter focuses on describing the concept of the visualization tool's design. First, functional requirements defined in Chapter 3 are distributed among two available system views. Then, the usage of the individual features is presented in the user scenarios. After that, the user flow is analyzed, upon which the system screens are defined for the future design. Finally, the possible states of the machines in the production hall are determined by analyzing the flow of events during the day in production.

#### 4.1 Features distribution

Multiple manufacturing companies that took part in the user research have had a TV screen at the production hall with an overview of the current production state. It would be wrong to deprive the production of such a helpful tool. That is why the final visualization tool will consist of 2 views: a TV view and an app view. The features will be distributed as shown in Table 4.1.

Implementation-wise, there will be only one app that will offer a view suitable for the TV screen. That view will display the current production schedule. When the user wants to interact with the visualization that he sees on the TV screen, he will grab a mobile device that will offer a more comprehensive range of features.

#### 4.2 User scenarios

Unlike problem scenarios, user scenarios describe the solution to the defined problem[23]. They consider the final software solution and intend to show how it contributes to the user's problem scenarios. It is good to use the Adam persona created before and describe how the system would help him overcome the problems he experiences. The user scenarios must also consider a context in which the system will be used, such as the current stage of production, Adam's location during the usage, sometimes even Adam's mood at the moment.

Production visualization tool	
TV	App
<ul style="list-style-type: none"> <li>■ Production overview</li> </ul>	<ul style="list-style-type: none"> <li>■ Production overview</li> <li>■ Interactive timeline</li> <li>■ Machine detail</li> <li>■ Notifications</li> <li>■ Calendar</li> <li>■ List of problems</li> </ul>

**Table 4.1:** Distribution of system features among two views.

#### ■ 4.2.1 Production delay

Adam walks through the production hall, where several machines operate on their tasks. Adam looks at the TV screen, where he sees an overview of the current production state. Immediately he notices that the currently executed order is delayed because of one of the machines. He grabs the tablet that he carries around to seek the cause of the delay in the app. Using an interactive timeline, he finds the exact moment when the delay began. He sees that the machine lacks the material. Adam calls the delivery company and discovers that the material will be delivered in about 2 hours. Thanks to the app, he can recalculate the production schedule that will be optimal regarding the situation. Order execution is back on track, whereas Adam is fully aware of the cause of the delay and can communicate with the customer about it. Nobody has to stay at work late.



**Figure 4.1:** Storyboard of production delay user scenario

### 4.2.2 Machine breakage

Adam sits at his office table in front of the computer. His tablet chimes as it receives a notification. The app informs Adam that a worker added a new problem regarding the packaging machine. Adam looks into details and sees that it requires a technician, who is also a user of the app. Adam assigns the problem to the technician, which changes the problem's state to "In progress." Adam knows that the technician will get a notification and handle the problem. The broken machine does not have to process any orders, so there is no need to change the schedule.

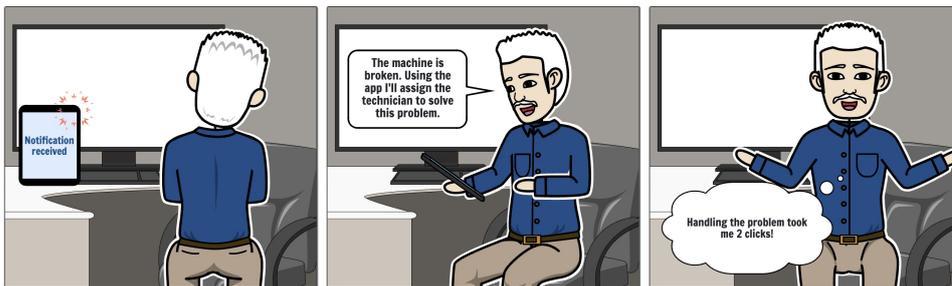


Figure 4.2: Storyboard of machine breakage user scenario

### 4.2.3 Catching up on events

Today is the first day at work after Adam's weeklong holiday in the mountains with his family. While he drinks his morning coffee, he opens the app to catch up on the last events in the production. In his notifications, he discovers a couple of problems that occurred during his time-off that were already resolved. He also sees that an order for a big hotel that buys Soap&Washes company's products was completed and shipped with a 1-day delay due to machine breakage. He opens his daily plan and sees a planned quality control for another order and awaiting service on the packaging machine. Now he can go to work knowing what to do.



Figure 4.3: Storyboard of catching up on events user scenario

### 4.3 User flow and system screens

The path undertaken by a typical user on an app to complete a task is referred to as user flow[24]. The user flow supports the design process by uncovering all the screens and UI<sup>1</sup> elements that have to be designed for the user to achieve his goals. Figure 4.4 describes all possible path the user of the visualization tool could follow. Blue parallelograms indicate app screens or functionalities, gray rectangles represent user actions, and yellow rhombuses symbolize "yes"/"no" user's decisions. Black dots indicate start and end points of the user's paths.

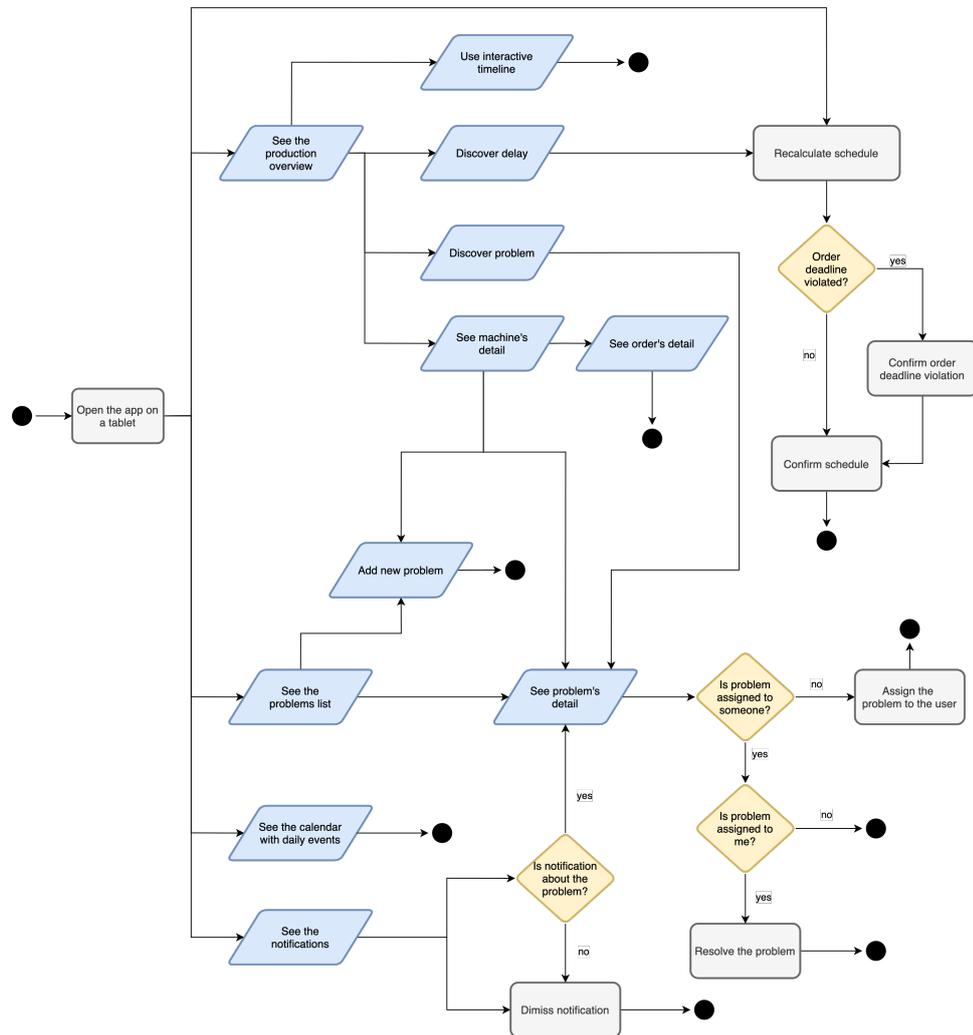


Figure 4.4: User flow chart.

<sup>1</sup>User Interface

Based on the user flow, it is possible to design the system's screens and the UI elements. The system will consist of nine views:

1. Production overview, which will variate between:
  - Production overview to display on the TV
  - Production overview with UI controls to interact with it through the app
2. Machine detail view
3. Order detail view
4. Problem list view
5. Problem detail view
6. A view for adding a new problem
7. Notification list view
8. Daily events list view
9. A dialog for schedule recalculation

## 4.4 Machine's states definition

The production overview consists of the machines placed according to their actual positions in the factory. This view displays the state of the machines and the progress of the orders being executed in the selected moment. To design the production overview, one should analyze in which states each machine could be. The state of a machine varies between being on and off throughout time as shown in Figure 4.5.



**Figure 4.5:** Machine state timeline.

The “on” state means that the machine operates on some task, whereas the “off” state describes the situation when the machine awaits a task. However, other entities also change their state in time and affect the machine state. One of these entities is the schedule, which state varies throughout the time as shown in Figure 4.6.

#### 4. Design Concept



**Figure 4.6:** Schedule state timeline.

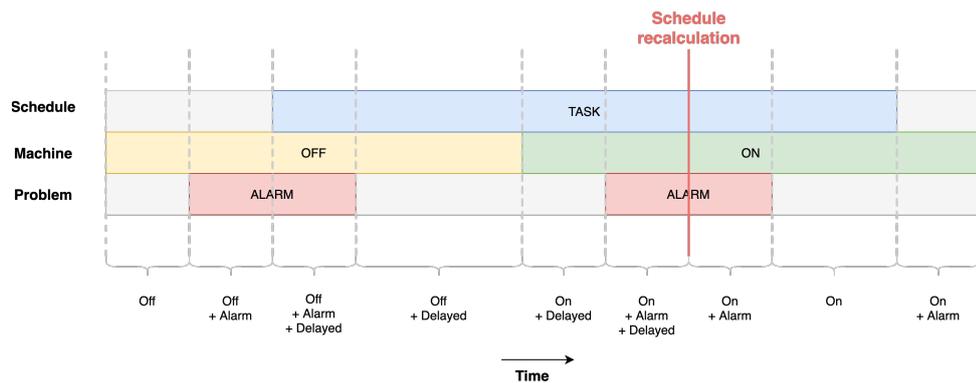
The schedule can either consist of some task for the machine to operate on, or not. Another entity that affects the machine's state is the problem. It either exists (and therefore, there is an alarm for the user) or not (see Figure 4.7). The possible problems with a machine could be for example:

- machine breakage
- material shortage
- increased scrap rate - determined by the analysis of the machine performance
- another - i.e., caused by a human factor



**Figure 4.7:** Problem state timeline.

When the state of the machine is put next to its corresponding state of schedule and state of a problem, several states can be distinguished at the spots where either one of the entities' states changes as shown in Figure 4.8.



**Figure 4.8:** Machine state timeline combined with the states of other entities.

The figure above can be interpreted as a following flow of events during the day at the production presented in Table 4.2:

Event	Machine state
Machine is <b>off</b> , waiting for a task to begin.	Off
Operator notices that there is something wrong with the machine and creates a new <b>problem</b> in the system.	Off + Alarm
Technician could not fix the machine in time. The machine is still <b>off</b> when the task in the schedule begins. As time goes by, the machine gets more <b>delayed</b> .	Off + Alarm + Delayed
Technician fixes the problem. The machine is still <b>off</b> and <b>delayed</b> .	Off + Delayed
Operator turns <b>on</b> the machine, so it begins working on a task in the schedule, but it is still <b>delayed</b> .	On + Delayed
Material is running short, so the operator adds a new <b>problem</b> to the system.	On + Alarm + Delayed
Production manager (Adam) resolves the delay by <b>recalculating the schedule</b> with the respect of expected time of the material delivery.	On + Alarm
Material is supplied to the machine. The machine is running <b>on</b> a task without a problem.	On
Machine still runs then there is no task in the schedule. System notifies the production manager with an alarm.	On + Alarm

**Table 4.2:** Flow of events connected to the changes in machine state.

## 4.5 Conclusion

Thanks to the future design concept, it is clear which UI elements must be designed to satisfy all the functional requirements and deliver the end-user the necessary features. Chapter 5 uses the concept to design the defined machine's states and the other UI elements in the system.



## Chapter 5

# User Interface Design

This chapter describes the process of the system design based on the functional requirements determined by the user research and the concept presented in Chapter 4. The design's starting point is the future solution mock-up presented in Figure 2.4. The process starts with designing the machine's element to display the machine's different states. The element is then integrated into the production overview design, consisting of other UI elements that help interact with the overview. Then, the design of the other system screens is described. The design of the individual features can be found in Appendix B.

### 5.1 Machine's states design

To display the state of the machine in the production overview, eight machine states must be designed. Each of these states must have two variants:

1. **Overview** - to display the most important information about the machines in the production overview.
2. **Detail** - to show additional information about the machine state, its progress on a task and occupancy during the selected period.

By zooming in or out in the production overview, the user will switch between seeing general and the most important information about the machines to seeing the detailed view of the single machine. The differences between displayed information for each state are described in Table 7.1.

After multiple iterations, the circle shape turned out to be the most suitable to represent the ongoing process on the machine and its completion state. The UI element of the machine will consist of 3 parts as shown in Figure 5.1:

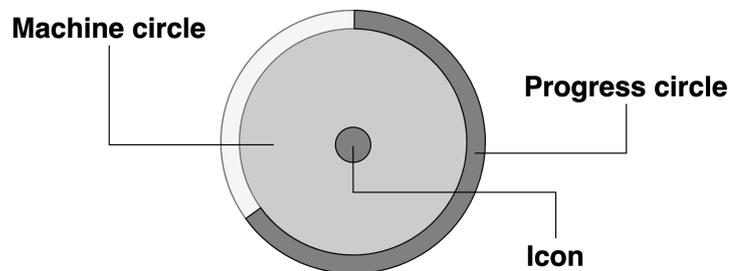
1. **Machine circle** - to display the machine "on"/"off" state.
2. **Progress circle** - to display the progress of the currently executed order and possible delay.

3. **Icon** - to indicate the problem with the machine, or support the “on”/”off” state.

Designed states, including their overview and detailed versions, are described in the following sections.

	Overview	Detail
<b>Platform</b>	TV, App	App
<b>Zoom state</b>	zoomed-out	zoomed-in
<b>Machines</b>	n	1
<b>Features</b>	Machine state Order progress	Machine detailed state Order detailed progress Machine name Machine occupancy Call-To-Action buttons

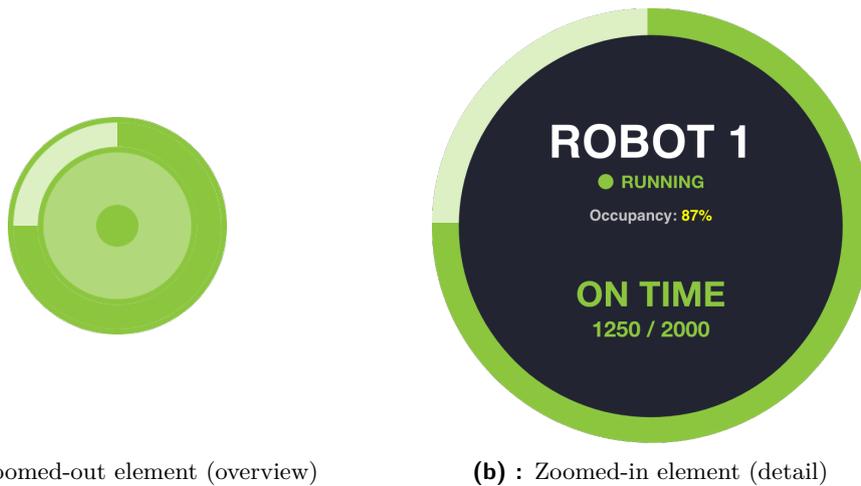
**Table 5.1:** Visible features in overview and detail of machine state



**Figure 5.1:** Scheme of machine UI element.

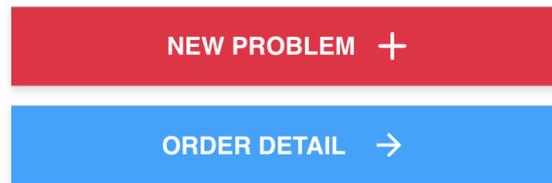
### 5.1.1 Machine is on

The design of the running machine is shown on Figure 5.2. The green color of the machine’s overview circle indicates the “on” state of the machine. The green color of the progress circle indicates the readiness of the currently executed task. After zooming in on the machine overview, the user will see a detailed machine view. This view will offer additional information about how many pieces of the product are produced, the machine’s name, and how much it was occupied in the selected period.



**Figure 5.2:** Design of "on" machine state.

Zoomed-in view also presents the user with the CTA<sup>1</sup> buttons, which allow him to add a new problem with the machine or see the detail of the currently executed order. The buttons are placed next to the machine element and are shown in Figure 5.3.



**Figure 5.3:** CTA buttons available in zoomed-in machine view.

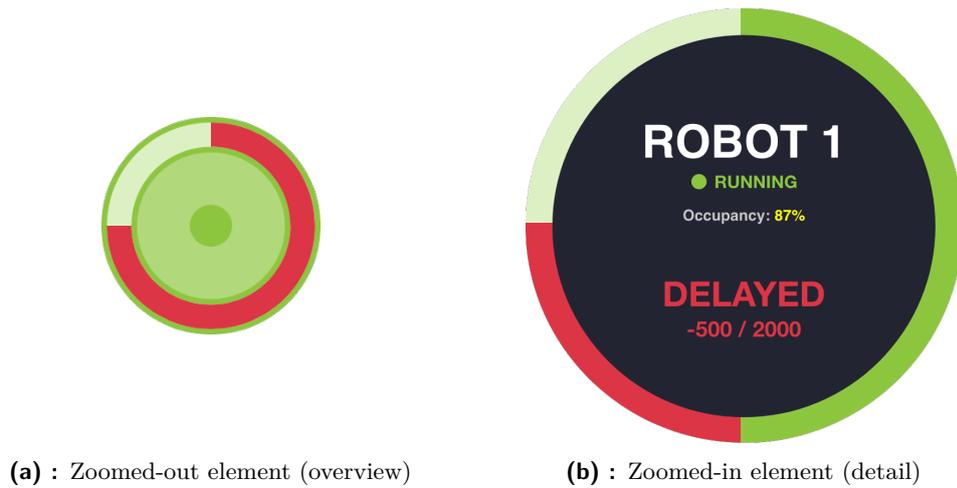
### ■ 5.1.2 Machine is on and delayed

When the currently produced amount of product is less than the expected amount according to the plan, it is crucial to let the user know about the delay. This state is presented in Figure 5.4. An overview view of the machine indicates the delay by the red color of the progress circle. In the detailed view, the user is presented with detailed information about the delay:

- the moment during the production in which delay has started to appear,
- the shortage of the produced product compared to the plan.

---

<sup>1</sup>Call-To-Action

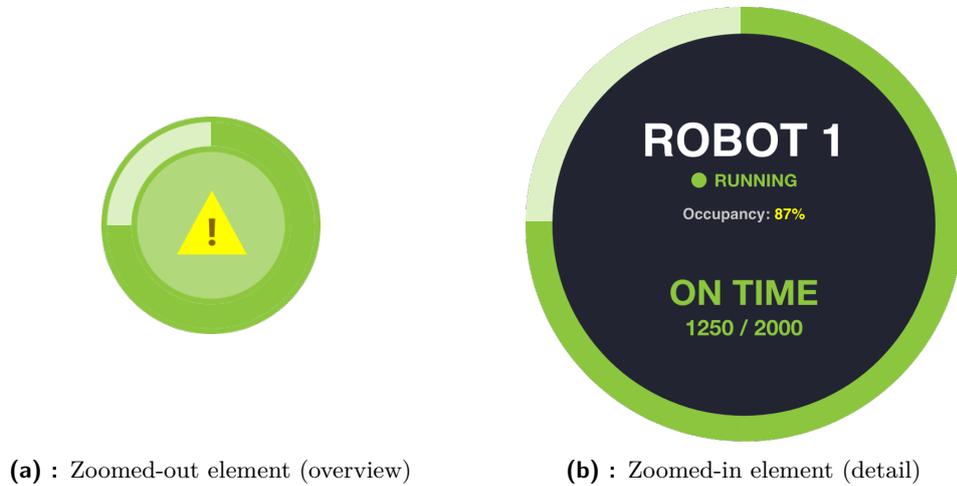


**Figure 5.4:** Design of "on + delayed" machine state.

In the detailed view, the CTA buttons shown in Figure 5.3 are available.

### ■ 5.1.3 Machine is on and has an alarm

The alarm is indicated by the warning sign in the center of the machine circle (see Figure 5.5).



**Figure 5.5:** Design of "on + alarm" machine state.

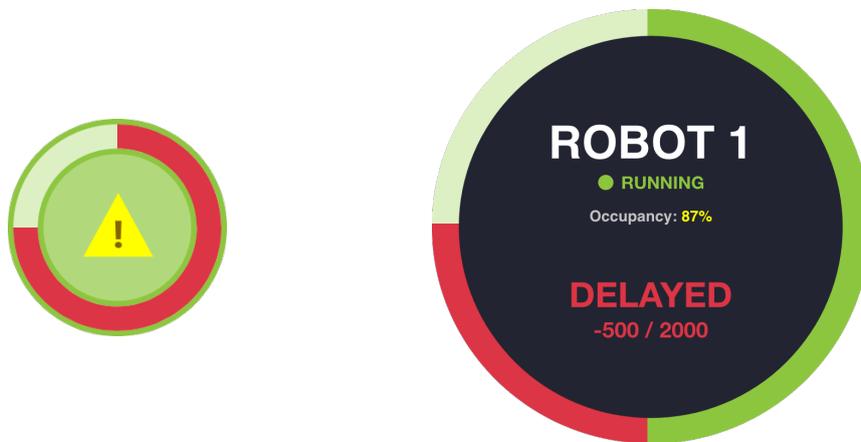
After zooming in, the user is presented with the list of the last 2 problems related to the viewed machine. The list is presented in Figure 5.6 and placed next to the machine's element. The user can see the list of all machine problems, add a new one, or see the order detail by using the CTA buttons shown in Figure 5.3.



**Figure 5.6:** List of problems available in zoomed-in machine view.

#### ■ 5.1.4 Machine is on, delayed and has an alarm

This state is a combination of the 3 states (see Figure 5.7). The “on” state is indicated by the green color of the machine’s overview circle, the “delayed” state is visualized by the red color of the order progress circle, whereas the “alarm” is indicated by the warning sign inside the machine’s circle. The detailed view is also a merge between 2 machine’s states which offers the list of the last 2 problems related to the viewed machine shown in Figure 5.6.



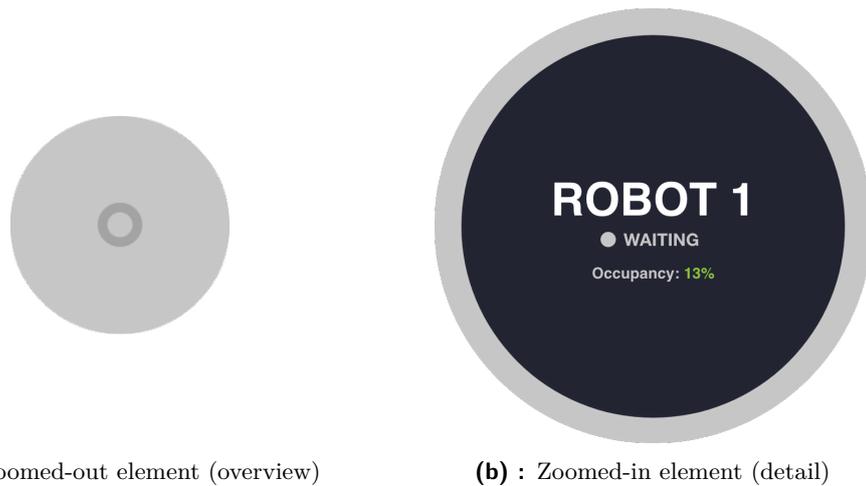
(a) : Zoomed-out element (overview)

(b) : Zoomed-in element (detail)

**Figure 5.7:** Design of "on + alarm + delayed" machine state.

#### ■ 5.1.5 Machine is off

The grey color of the machine circle shown in Figure 5.8 indicates the “off” state of the machine. The detailed view shows the machine’s name, a label of its state, occupancy and offers a CTA to add a new problem with the machine.

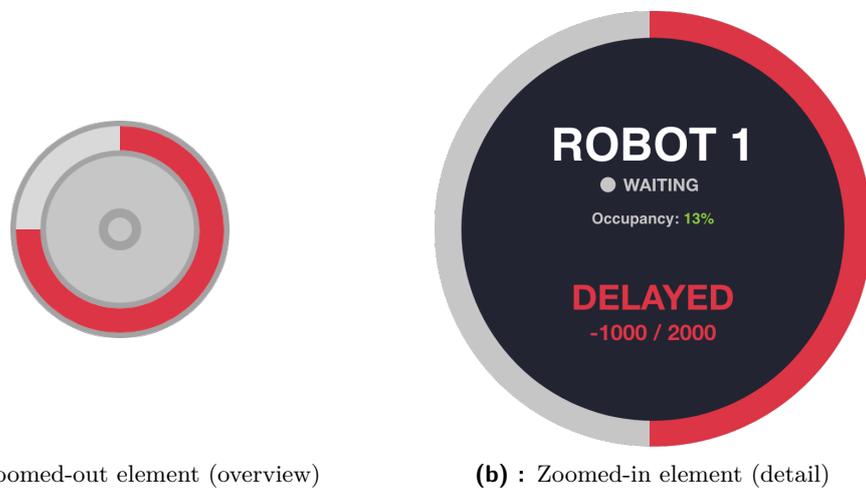


(a) : Zoomed-out element (overview)      (b) : Zoomed-in element (detail)

**Figure 5.8:** Design of "off" machine state.

### ■ 5.1.6 Machine is off and delayed

The delay of the waiting machine is indicated similarly to the running machine — by the red color of the order progress circle (see Figure 5.9). The detailed view shows the shortage of the product compared to the plan. CTA buttons shown in Figure 5.3 allow the user to add a new problem or explore which order is delayed on the viewed machine.

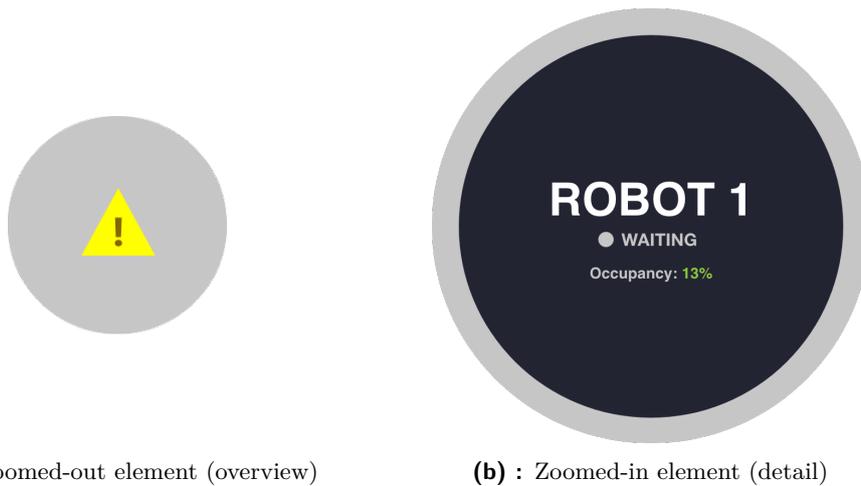


(a) : Zoomed-out element (overview)      (b) : Zoomed-in element (detail)

**Figure 5.9:** Design of "off + delayed" machine state.

### ■ 5.1.7 Machine is off and has an alarm

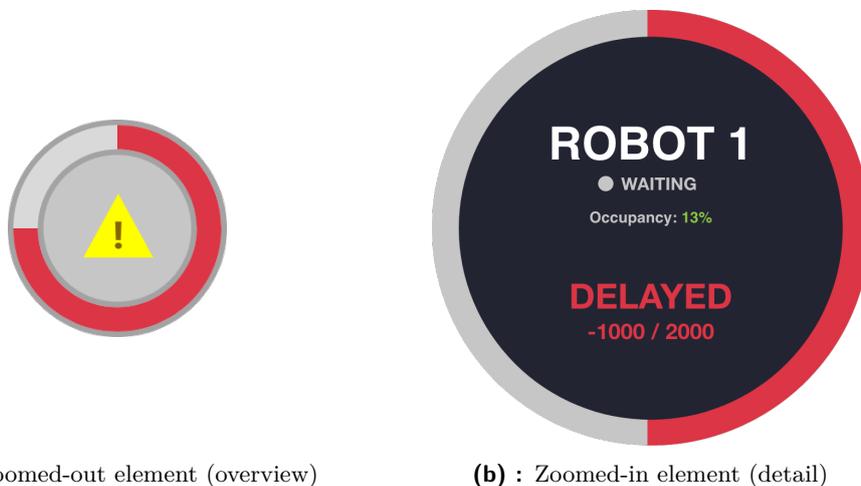
The warning icon in the middle of the machine's circle indicates some problem with the washing machine (see Figure 5.10). The detailed view offers a list of the last 2 problems, as shown in Figure 5.6, and a possibility to see the rest of them on a separate view.



**Figure 5.10:** Design of "off + alarm" machine state.

### ■ 5.1.8 Machine is off, delayed and has an alarm

The state shown in Figure 5.11 is a combination of the 3 states. The grey color of the machine circle indicates the “off” state of the machine, the red color of the task progress represents the “delay”, and the warning icon indicates the “alarm” about the machine’s problem. The detailed view is similar to the corresponding view of the running machine except for the grey color and the “Waiting” state label.

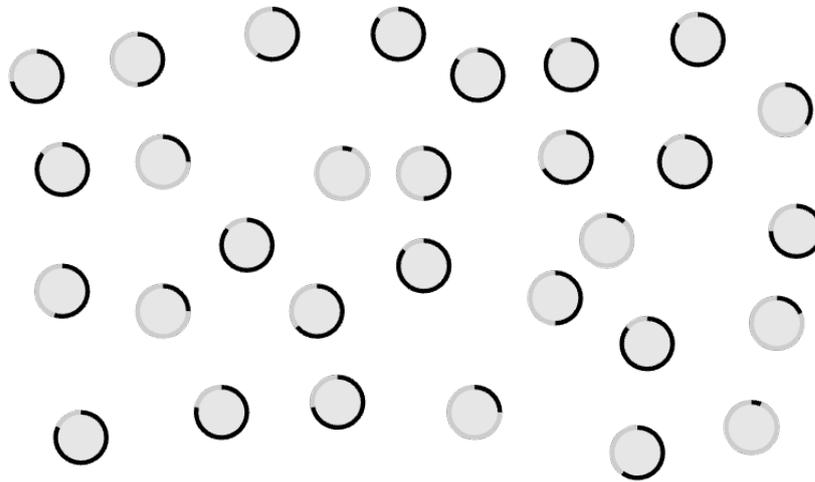


**Figure 5.11:** Design of "off + alarm + delayed" machine state.

## ■ 5.2 Production overview

The production overview takes the biggest part of the prototype and reflects the layout of the factory in real life. It is the only feature available for both TV and the app view. It consists of the machines in the production hall and

displays the production state in the chosen moment. Like existing systems, the visualization tool will offer a viewer information of every machine's state and its progress on the currently executed task. The Lo-Fi idea of the production overview is shown in Figure 5.12.



**Figure 5.12:** Lo-Fi idea of the production overview.

The Hi-Fi design consists of the designed machine states as well as the control panel to interact with the production overview. Figure 5.13 below shows:

- **Menu** - to switch between parts of the system. The menu consists of:
  - The avatar and the name of the logged-in user
  - Calendar button to display daily events
  - Warning button to display the list of the problems
  - Recalculate schedule button - to create a new schedule when a delay appears
  - Bell button to display the list of notifications
  - Log off button to log out the user from the system.
- **Period control** - to change the range of the interactive timeline.
- **TV view toggle** - to switch between a view for the TV and app.
- **Production overview** - the view can be zoomed-in to display the machine's detail.
- **Interactive timeline** - moving the pin of the timeline will affect the currently displayed state of the machines and orders.



**Figure 5.13:** Hi-Fi prototype of the production overview with labeled parts.

Each machine has a rectangle behind it to help the user connect the displayed circles to the actual machines in the factory. The view also resembles the walking paths between the machines. These elements are intentionally semi-transparent to make the machine circle elements appear more apparent.

By clicking on the TV view, the user will switch between omitted production overview for the TV screen and the overview with the controls dedicated to the usage in the app. The example of the TV is shown in Figure B.1. This view omits the menu, period control, and the interactive timeline, leaving the production overview and the toggle to be the only elements displayed on the screen.

By switching back to the app view, the user can zoom in production overview to some of the machines, which will switch the machine's element to appear as the detail of the machine. This situation is shown in Figure B.3.

The user can move the pin of the interactive timeline to see how the machine's state changed over time and how the order progressed. This feature could come in handy when there is a need to detect the exact moment when the order started to gain on delay.

## 5.3 Order detail

When there is a task in the schedule for the machine, the system allows the user to see the order's details by clicking on one of the CTA buttons after zooming in. Order detail opens on the right side of the screen and presents the user with additional information about the order available in the system (see Figure B.4).

## 5.4 Daily events

By clicking on the calendar button, the system opens the panel on the right side of the screen that shows the events in production that day (see Figure B.5).

These events are given by the production schedule and can be of such natures:

- Reminder to check on the material before the production starts
- Reminder to check on the ready product
- Reminder to order more material (can be implemented when material management functionality is added to the system)

The user can switch the displayed day by clicking on the arrows next to the “Today” label on the top of the list.

## 5.5 Problem management

Clicking on the warning button in the menu will open the panel on the right side of the screen (see Figure B.6). In this panel, the user can see the list of all of the problems in the system. A problem can be manually added by some user or automatically detected by the system (i.e., increased scrape rate of the machine).

The system allows the user to filter the problems by the machine it appears on, the user to whom the problem is assigned, and the state of the problem. Possible states of the problem are:

- Open
- In progress
- Closed

The state diagram of the problem is shown in Figure 5.14.

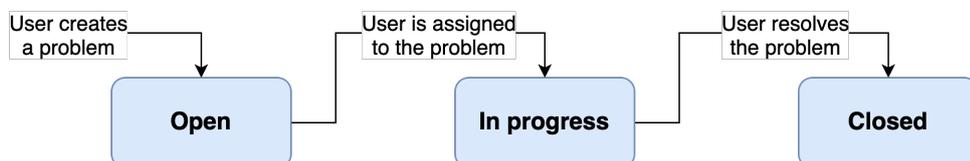


Figure 5.14: Problem state diagram.

By clicking on the “New +” button, the user is redirected to the new problem form shown in Figure B.7. The user can specify which machine is

affected by the problem, describe a problem by a particular type, assign the problem to another user of the system and add additional information to the problem's description. Available problem types are:

- Breakage
- Material shortage
- Planned down-time
- Increased scrap rate
- Other

After the problem is added, the user is redirected to the problem detail presented in Figure B.8. On this screen, the user can change the state of the problem or delete it.

## ■ 5.6 Notifications

Thanks to notifications, the user can keep up with the events at the factory. The notification has 3 levels:

- **Success** - information about successful start or end of the order production.
- **Warning** - information about new problems with the machines.
- **Danger** - information about the late start or end of the order production, which can cause the delay.

The list of notifications is available by clicking on the bell button in the menu and is shown on the right side of the screen, as shown in Figure B.9.

Notifications can be dismissed by clicking on the cross icon next to them, or in case of the problems' notification, can be clicked on, which will redirect users to the problem's detail.

## ■ 5.7 Schedule recalculation

If any of the machines is delayed, the system offers the user to recalculate the schedule respectfully to that delay. The button to do so is always present in the menu on the left side of the system and leads the user to the schedule recalculation dialog composed of 2 steps shown in Figure B.10 and Figure B.11 accordingly.

The dialog shows how many steps are ahead of the user. The first step lets the user choose the desired start of the new schedule if there is a need to wait for the material or the working running late to set up the machine. The second step alerts the user of the possible violation of the order deadline. Thanks to the optimization algorithm, the calculated schedule is always optimal, even when a deadline violation happens. The user can then choose whether to apply the new schedule or not.

## 5.8 Design testing and discussion

According to the User-Centered-Design principles, the created design must be evaluated with the users. Since the design was created with Adobe XD application[25], it is convenient to use it for prototype creation as well. Adobe XD allows the designer to set up the change of screens upon the user action. The prototype used the "tap" action to simulate the functionality of clicking the menu and Call-To-Action buttons and the "drag" action to simulate the zooming-in of the production overview. Thanks to an Adobe XD application on the tablet, the prototype could be tested as it was the real visualization tool running as a tablet app. The usage of the prototype can also be recorded, which allows the designer to play the user's interaction with the prototype even after the testing is finished. The goals of the design testing were:

- Test if the states of the machines are recognizable.
- Test if there is a pop-out effect for the states that require the user's attention.
- Test if the placement of the element is convenient for tablet usage.
- Test if the sizes of the machines are convenient for the TV display.

The test participants were given the following tasks:

- Look at the TV screen with the product overview. Which machines are on and which are off? Which machines are delayed? Which machines have some kind of a problem?
- Use a tablet to recalculate the schedule.
- See the list of notifications.
- Explore the list of all problems in the factory.
- Look at your daily plan.
- Move your finger to zoom in to the machine's detail. Add a new problem to the zoomed-in machine.

Participants could express their impression of the design as they were testing it. All ideas and suggestions for improvement are presented in conclusion below. After the testing process was completed, there was a discussion with the participants about other possible features of the visualization tool. Participants' ideas were taken into account and projected into the functional requirements described in Chapter 3.

### ■ 5.8.1 Conclusion

The test showed that designed machine states were recognizable, and the pop-out effect was present for the states that needed user's attention. The sizing was also suitable for the TV screen since no participant had a problem describing the state correctly, even further from the screen. There were some suggestions for the user interface of the system. Some of them come directly from the users, and some come from the designer's observation of how the system is used. The changes to be implemented in the next stage of the development are:

- Move the menu to the right side of the screen.
- Highlight the "Recalculate schedule" button when there is a delay.
- Move the notifications button to be the first one in the menu.
- Add machine names to the zoomed-out production overview.
- Display the running time of the machine in the machine's detailed view.
- Restrict the application usage to the horizontal mode.

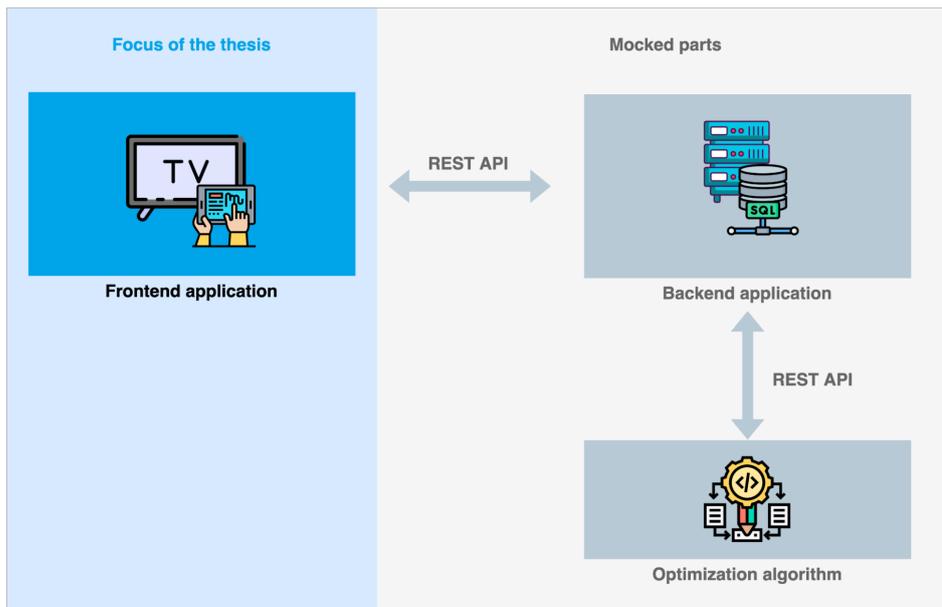
The testing of the created design was an overall success, which means that the development process can proceed to the implementation stage.



## Chapter 6

### Implementation

This chapter describes the implementation stage of the production visualization development. The goal is to implement the design described in Chapter 5 as a frontend application that displays mocked data from the backend. Figure 6.1 shows the exemplary architecture of the production visualization tool and highlights the focus of this thesis that will be implemented.



**Figure 6.1:** System architecture with highlighted focus of the thesis.

There have to be 3 parts of the system for it to function correctly:

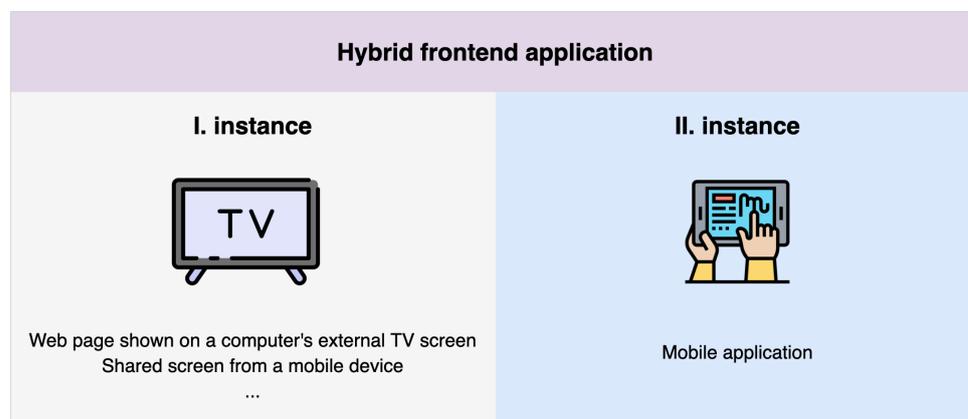
- **Frontend application** - is an application that is used directly by the user. It presents the user with relevant information about the schedule and events in the production and allows him to interact with that data. The application communicates with the system's backend part through the REST API.



**PWA.** Progressive Web Apps[27] are implemented as standard web pages that run in a standalone window instead of a browser's tab. They could be launched from the home screen on mobile devices like any other installed application. The most significant benefit of using PWA is having a single codebase for every platform. The downside is that the PWA is not fully supported on iOS[28], which happens to be the most used platform.

**Hybrid.** Hybrid mobile applications[29] take the best from the native and PWA applications. They are implemented as a single application and then compiled into the native code of the target platform. The only downside of the hybrid applications is the need to use different libraries and frameworks such as Flutter or Ionic and third-party plugins to access the device's native features, such as camera, Bluetooth, notifications, and others. It makes hybrid apps slower and less robust compared to native applications.

Given that the frontend application must play a presentational and interactive role, a web application that can run on a mobile device seems like the right choice. Hybrid applications have an advantage in iOS support, making this method the most beneficial for the production visualization implementation. The usage of the application will require having two instances of it: one for the presentation (i.e., a web page running on the computer that uses a TV screen as an external monitor), and one for the interaction (same application running on a mobile device). The usage example is presented in Figure 6.2.



**Figure 6.2:** Example of the production visualization hybrid application usage.

### 6.1.2 Hybrid development

There are plenty of frameworks to choose from when it comes to hybrid application development. Each allows the developer to write a single application and build it for multiple platforms. Three of these frameworks have gained the sympathy among developers more than the others, making them a candidate to choose from for the implementation phase of this thesis.



frameworks would be in place. Like other concurrent technologies, each framework has its pros and cons, but one feature is common for all of them. The code is separated into components, which behavior is controlled by the input parameters. Components are reusable and tend to encapsulate a specific functionality or UI element. Each framework, however, tends to treat its components differently.

**Angular.**[38] The Angular framework was released in 2010 by Google making it the oldest framework of the lot. It used a Typescript language[39] and implemented a strong MVC (Model-View-Controller) pattern. In later versions of Angular, the association with MV\*-patterns became less strict, but the separation between the component's view from its logic persisted. Angular components use services to handle the application's business logic or fetch the data. An Angular component must be declared in a module, representing a container for the block of code dedicated to a specific domain or workflow. The usage of services, modules, and components makes an Angular application appear more structured.

**React.**[40] The Facebook released React framework in 2013 and has been actively using it in its products like Instagram and WhatsApp. Unlike Angular, React combines the logic and view of its components. The framework uses the JSX syntax that allows elements to contain HTML and JavaScript at the same time. React is less strict regarding the project structure, giving the developer more freedom to create the application architecture himself.

**Vue.**[41] The Vue framework was created by the ex-Google employee in 2014. The MVVM (Model-View-ViewModel) model partly inspired its design. Components in Vue tend to be small and are referred to as SFC (Single File Components) containing HTML, JavaScript, and CSS in one file. The Vue framework is more lightweight than React and Angular, which downsides in need for additional plugins for advanced functionalities.

The Angular framework was the most suitable choice for implementing the production visualization tool in the Ionic framework. Its precise project structure and required typings will make the frontend application more readable and easy to maintain. The separation of the components' logic from their UI and organizing business logic into services will make integrating the frontend part into a complex system more convenient.

#### ■ 6.1.4 Selected technology

After analyzing multiple ways to implement the designed visualization tool, the most convenient one was creating a hybrid application with the Ionic framework that uses Angular. The implementation described in this thesis uses the latest stable versions of both frameworks: Ionic 6.0.1 and Angular 13.1.1. The manual for installing and running the implemented frontend application is presented in Appendix E.



```
1 export interface Machine extends LayoutElement {
2   id: string;
3   name: string;
4   occupancy: number;
5   orders: Order[];
6   problems: Problem[];
7 }
```

The Machine interface contains:

- **id** - a string value of the machine's id in the database;
- **name** - a string value representing the name of the machine;
- **occupancy** - a number value representing the machine's occupancy during the selected period. This value is obtained from the mocked backend, so the frontend application does not need to worry about occupancy calculation for each machine. It just sends a request with the period selected by the user and waits for the backend to send the calculated occupancy in the machine's object.
- **orders** - a list of orders that will be executed during the calculated schedule of the machine;
- **problems** - a list of problems assigned to the machine;

Apart from its properties, the Machine interface extends the LayoutElement interface discussed further in Section 6.3. The Machine class, in addition to Machine interface properties, offers methods to handle different machine states:

- **isOn(date?: Date)** - a method that returns a boolean value representing whether the machine is on or off;
- **isDelayed(date?: Date)** - a method that returns a boolean value representing whether the machine is delayed on some order or not;
- **getStartedOrder(date?: Date)** - a method that returns an order that was started and may or may not has been finished;
- **getScheduledOrder(date?: Date)** - a method that returns an order which was supposed to start according to the calculated schedule but is not yet supposed to end;
- **getProblems(date?: Date)** - a method that return a list of the problems assigned to the machine.

Each method takes a date as a parameter. This feature comes in handy when the user interacts with the timeline exploring the machine's state in the past or future. When the date parameter is not defined, the current date is used instead.

### 6.2.3 Order

The customer's order executed on one of the machines is defined by the following interface:

```

1  export interface Order {
2    id: string;
3
4    // User data
5    orderNumber: string;
6    productCode: string;
7    requiredAmount: number;
8    requiredEnd: Date;
9
10   // Schedule
11   plannedStart: Date;
12   plannedEnd: Date;
13
14   // Reality
15   start: Date;
16   end: Date;
17   producedAmount: number;
18   currentState: OrderState;
19 }

```

The data entered into the system by the user is defined by the properties `orderNumber`, `productCode`, `requiredAmount`, and `requiredEnd`. The calculated data by the scheduling algorithms are represented in the properties: `plannedStart` and `plannedEnd`.

The backend reads the real-time data of the machines in the production hall and provides the frontend with the following properties of the Order interface: `start`, `end`, `producedAmount` and `currentState`.

Simillary to the Machine, the Order class offers additional methods:

- `isDelayed()` - returns a boolean value representing if the order's state is equal to the enum value `OrderState.Delayed`;
- `predictProducedAmountForDate()` - returns an integer value of the product amount to be produced in the future point of the schedule.
- `getProgressPercentage(date?: Date)` - converts the value of produced amount to the percentage value due to the method's date parameter.

### 6.2.4 Problem

A problem interface describes a machine's problem stored in the system. Some of the problems could be detected automatically by analysing the machine's performance (i.e., increased scrap rate), or could be entered manually by the user. The Problem interface has the following form:

```

1 export interface Problem {
2   id: string;
3   state: ProblemState;
4   machineId: string;
5   problemType: ProblemType;
6   createdOn: Date;
7   responsiblePerson?: User;
8   resolvedOn?: Date;
9   createdBy?: User;
10  resolvedBy?: User;
11  note?: string;
12 }

```

The Problem interface defines the problem's state (`Open`, `InProgress`, `Closed`), its type (`Breakage`, `IncreasedScrapRate`, `PlannedDownTime`, `MaterialShortage`, `Other`), machine and a user is was assigned to, information about who and when created the problem, occasionally some comment.

## 6.3 Production overview

Production overview takes the biggest area of the frontend application (see Figure 5.13) since it visualizes the production process. It must offer several functionalities:

- **Switching between TV and app view.** The TV view will display machine elements bigger and hide all interactive features except the button to switch back to the app view. Practically, a web instance will be switched to the maximized view for the TV screen, whereas a mobile instance will use the minimized view that allows interactions. The design of the TV view is presented in the figure, and the design of the app view is in the figure. Button "Maximize"/"Minimize" behaves as a toggle between two views.
- **Placing machines accordingly to their actual position in the production hall.** The production overview must fit all the machines engaged in the production, regardless of the screen size, and respect the machine's position and size assigned to it by the user.
- **Switching between overview and detail of the machine.** A frontend application must allow the user to zoom in to see the detailed machine view and zoom out to see the overview of all machines. Zoom actions must be executed upon pinch gesture. After zooming in, it has to be possible to move a zoomed-in view by a touch gesture.

Whereas switching between TV and app view is implemented upon toggle button click (see "TV view toggle" in Figure 5.13), machines' placement and interactions with the overview and switching between machines' states are a bit more complex to implement. Therefore, the way these functionalities were implemented will be described further in the following sections.

### 6.3.1 Machines' placement

The Machine interface described in Subsection 6.2.2 extends a LayoutElement interface, used for any elements in the production overview area. The LayoutElement interface and LayoutPosition interface have the following structure:

```

1 export interface LayoutElement {
2   position: LayoutPosition;
3   width: number;
4   height: number;
5 }

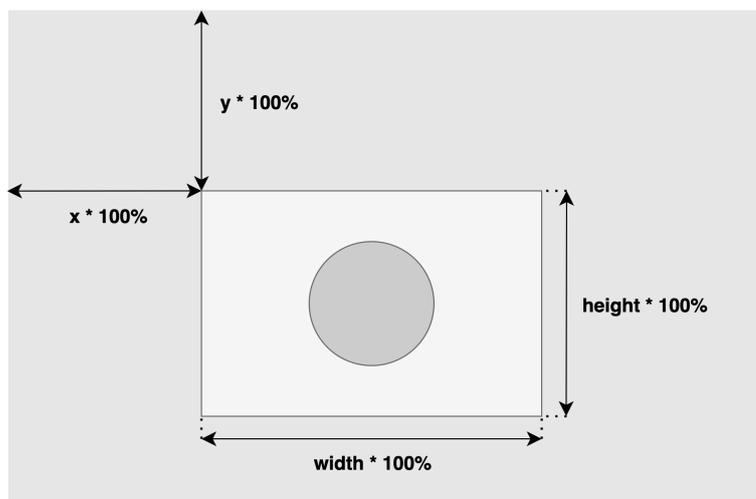
1 export interface LayoutPosition {
2   x: number;
3   y: number;
4 }

```

The `width` and `height` properties define the rectangle placed behind the machine circle element, which should resemble the actual size of the machine. The `position` property of the `LayoutPosition` interface defines a machine's placement in the production hall by defining two properties:

- `x` - the distance between the left side of the overview and the top left corner of the machine's rectangle;
- `y` - the distance between the top side of the overview and the top left corner of the machine's rectangle.

To make the production overview responsive for the different screen sizes (TV screens tend to be larger than a tablet screen) and the preserve the size and placement of the machine, the `width`, `height`, `x`, and `y` properties must be defined relatively as a decimal number between 0 and 1. An example of a machine's placement and size calculation is presented in Figure 6.3.



**Figure 6.3:** Calculation of the machine's placement in the production overview.

### 6.3.2 Switching between overview and detail

When all the machines engaged in production are placed in the production overview, it is time to implement the pan and zoom interactions. For this purpose, the project used the Panzoom library[44], which uses CSS transforms to add panning and zooming functionality to the element. It is compact (3.7kb gzipped) and does not add any significant weight to the application's final build. The library can detect a user's pinch gesture on the mobile platform and a mouse scroll event on the web platform, upon which the zoom effect is performed.

It is also essential to define the bound value for switching between displaying the overview and the detail of the machine's element. The application allows the user to enlarge the zoomed-out overview up to 4 times. Halfway through the process, when the scale is enlarged or reduced twice, the machine's element switches its view between the overview and the detail.

When styling both variants with CSS, making elements bigger than smaller is always easier. That is why the more complex design of the zoomed-in machine view will be styled as a normal-sized element (corresponding to the scale of 1). In contrast, the simpler zoomed-out machine view will have enlarged sizes and labels (corresponding to the scale of 0.25). Enlarged elements will be compensated by the smaller scale value, resulting in looking the same size as the elements in the zoomed-in view. The sizing of the element's contents is defined relatively to the `font-size` CSS property of the parent element. This way, setting the `font-size` value for the zoomed-out element four times larger than the `font-size` value of the zoomed-in element will be enough to even out the sizing. The transition logic between zoomed-out and zoomed-in design accordingly to the current scale of the production overview is presented in Figure 6.4.

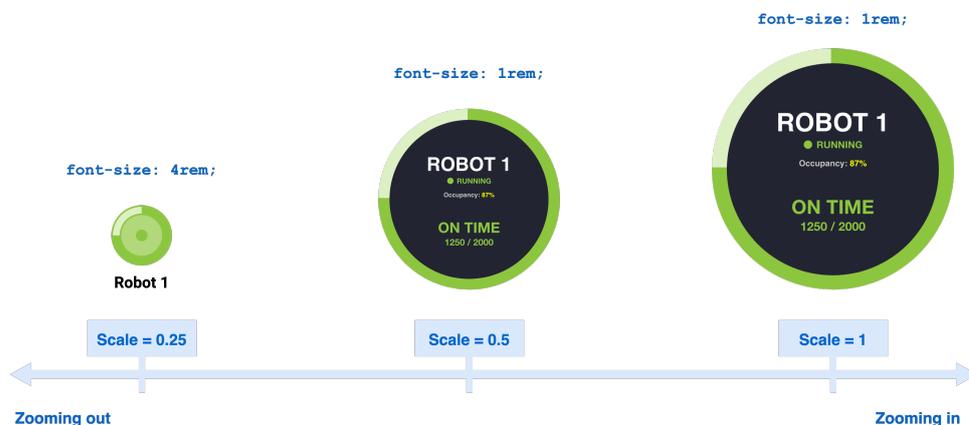


Figure 6.4: Approach for switching between overview and detailed machine view.



GET `/api/schedule/parameters` - returns available parameters for the new schedule calculation.

Response **ScheduleParametersDtoOut**:

```
1 {
2   "orders": [OrderDtoOut],
3   "machines": [MachineDtoOut]
4 }
```

POST `/api/schedule/recalculate` - calculates the new schedule according to the user's parameters.

Request body **ScheduleParametersDtoIn**:

```
1 {
2   "orderIds": [string],
3   "machineIds": [string],
4   "start": string
5 }
```

Response **ScheduleDtoOut**:

```
1 {
2   "start": string,
3   "end": string,
4   "calculatedOn": string,
5   "isDelayed": boolean,
6   "machines": [MachineDtoOut]
7 }
```

## 6.5.2 Machine endpoints

GET `/api/machine/:id` - returns the machine by the id.

Response **MachineDtoOut**:

```
1 {
2   "id": string,
3   "name": string,
4   "occupancy": number,
5   "orders": [OrderDtoOut],
6   "problems": [ProblemDtoOut],
7   "position": {
8     "x": number,
9     "y": number,
10  };
11  "width": number,
12  "height": number
13 }
```

GET `/api/machine` - returns the list of all machines in the system.

Response **MachineListDtoOut**:

```
1 {
2   "machines": [MachineDtoOut]
3 }
```

### 6.5.3 Order endpoints

GET /api/order/:id - returns the order by the id.

Response **OrderDtoOut**:

```
1 {
2   "id": string,
3   "orderNumber": string,
4   "productCode": string,
5   "requiredAmount": number,
6   "requiredEnd": string,
7   "plannedStart": string,
8   "plannedEnd": string,
9   "start": string,
10  "end": string,
11  "producedAmount": number,
12  "currentState": string
13 }
```

GET /api/order - returns the list of all orders in the current schedule.

Response **OrderListDtoOut**:

```
1 {
2   "orders": [OrderDtoOut]
3 }
```

### 6.5.4 Problem endpoints

GET /api/problem/:id - returns the problem by id.

Response **ProblemDtoOut**:

```
1 {
2   "id": string,
3   "state": string,
4   "machineId": string,
5   "problemType": string,
6   "createdOn": string,
7   "responsiblePersonId": string,
8   "resolvedOn": string,
9   "createdById": string,
10  "resolvedById": string,
11  "note": string
12 }
```

GET `/api/problem` - returns the list of all problems in the system.

Response [**ProblemListDtoOut**]:

```
1  [{
2    "problems": [ProblemDtoOut]
3  }]
```

POST `/api/problem/add` - creates a new problem in the system.

Request body **ProblemDtoIn**:

```
1  {
2    "machineId": string,
3    "problemType": string,
4    "createdOn": string,
5    "responsiblePersonId": string,
6    "resolvedOn": string,
7    "createdById": string,
8    "resolvedById": string,
9    "note": string
10 }
```

Response **ProblemCreatedDtoOut**:

```
1  {
2    "id": string,
3  }
```

PUT `/api/problem/:id` - modifies existing problem by its id.

Request body **ProblemEditDtoIn**:

```
1  {
2    "machineId": string,
3    "problemType": string,
4    "responsiblePersonId": string,
5    "resolvedOn": string,
6    "resolvedById": string,
7    "note": string
8  }
```

DELETE `/api/problem/:id` - deleted existing problem by its id.

### 6.5.5 Notification endpoints

GET `/api/notifications/:userId` - returns the list of unseen notification for the user.

Response [**NotificationDtoOut**]:

```
1  [{
2    "id": string,
3    "date": string,
```

```
4   "level": string,  
5   "text": string,  
6   "problemId": string  
7 }]
```

POST /api/notifications/:id - mark notification as seen.

### 6.5.6 Daily events endpoints

GET /api/events - returns the list of events in the factory for the given date.

Request body **EventDtoIn**:

```
1 {  
2   "date": string  
3 }
```

Response [**EventDtoOut**]:

```
1 [{  
2   "start": string,  
3   "end": string,  
4   "title": string,  
5   "level": string  
6 }]
```

# Chapter 7

## Usability Testing

The purpose of the test is to find usability issues of the system prior to its integration among the end-users. The test will:

- **evaluate** the visualization of the manufacturing process in terms of clarity of displayed machines states and usability of the interactive timeline,
- **determine** if the system supports the user in his everyday tasks.

To fulfill the purpose of the test 3 research questions must be answered:

- **Q1** - What usability issues does the system have?
- **Q2** - Is the visualization convenient for displaying the past and future production schedule?
- **Q3** - Would the user enjoy using the system during his workday?

Observing the participants during their interaction with the system and interviewing them afterward will reveal the answers to the research questions listed above.

### 7.1 Participants

The target group for the visualization tool is production managers whose responsibility is to take care of the production process and plan the execution of production tasks to satisfy an order's deadline. However, given how different were the companies in which user research participants worked, it could be distracting for them to test the system that is not designed precisely for their company structure and production processes. The main goal of usability testing is to test how pleasant the usage of the system is, not how well it fits a specific business. That is why most usability testing participants did not have any background in the manufacturing field. This choice also contributed to testing how easy it is to use the system for people who have not worked with similar tools before.

### 7.1.1 Screener

There are, however, some criteria for the candidates to be selected as participants in the usability testing. The screening method was used to select participants based on the following criteria:

- **Age** - the testing must be done with adults between 18 and 65 years old.
- **Usage of mobile devices** - a participant must own a smartphone and a tablet. This way, it is assured that participants are familiar with the concept of a mobile application and may operate it on a tablet.
- **Usage of mobile applications** - the visualization tool requires the user to be familiar with notifications, actions like zooming-in or zooming-out, and using such controls as range, menus, buttons, and dialogs in a mobile app. The goal is to filter out the candidates whose smartphone usage is limited to a few apps. To select the right participants, the map and calendar applications are chosen for the corresponding question in the screener. Candidates who use these apps at least once a month were selected for the usability testing of the visualization tool.

Participants were selected by the screener composed of the following questions:

Question	Answer options	Target
What is your age?	-	18 - 65
Do you have a smartphone?	Yes No	Yes
Do you have a tablet?	Yes No	$\geq 1$ yes $\geq 1$ no
How often do you use navigation on your phone or tablet?	Every day 2-3 times a week 2-3 times a month 2-3 times a year Never	Every day 2-3 times a week 2-3 times a month
How often do you use the calendar to write down events/tasks on your phone or tablet?	Every day 2-3 times a week 2-3 times a month 2-3 times a year Never	Every day 2-3 times a week 2-3 times a month

**Table 7.1:** Usability testing screener

## 7.1.2 Number of participants

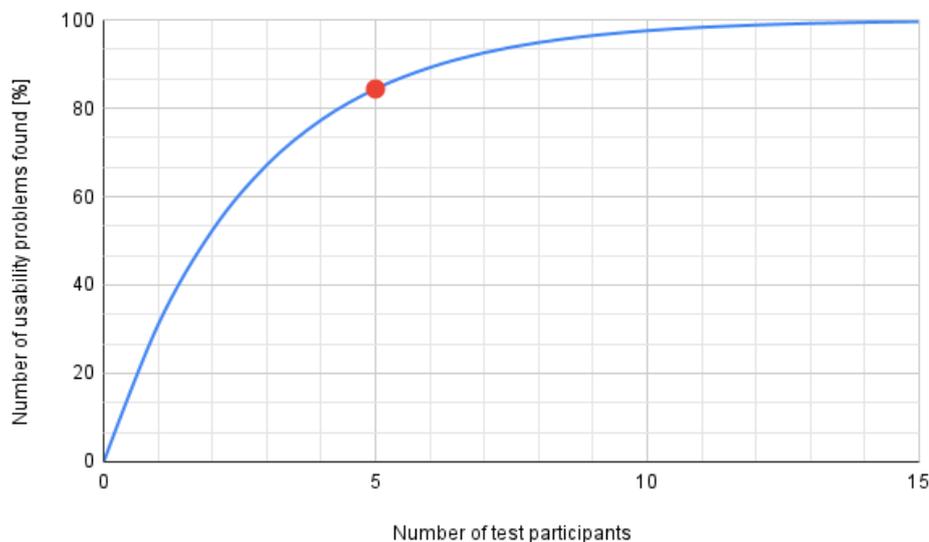
The research conducted in 1993 by Tom Landauer and Jakob Nielsen[49] revealed that the amount of problems found by usability testing conducted with  $n$  participants is:

$$N * (1 - (1 - L)^n)$$

where:

- **N** - is the total number of usability problems in the system,
- **L** - is the percentage of usability issues discovered while testing the system with a single user. Typical value of  $L$  is 31

Plotting the dependency between the number of discovered usability problems and  $n$  for  $L = 31$  gives the result presented in Figure 7.1.

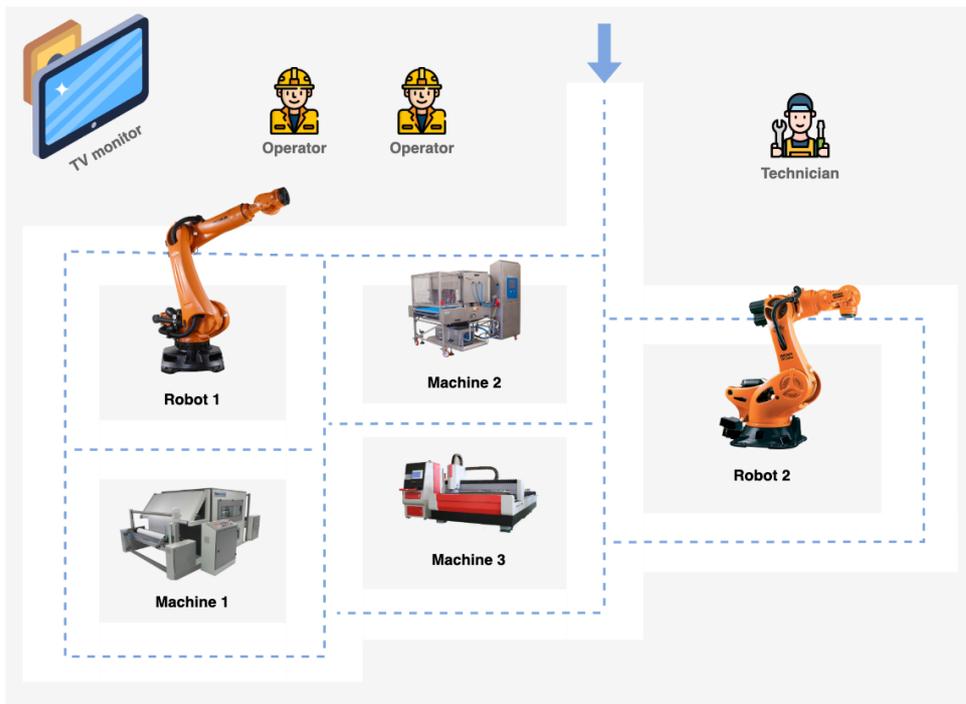


**Figure 7.1:** Number of discovered usability problems according to the number of test participants.

The figure above shows that the number of newly discovered usability issues stops after 15 participants. In other words, adding more users results in discovering the same problems repeatedly. The curve also shows that testing with only 5 users reveals 83% of usability issues. This ratio is satisfying for the goal of this thesis since the test is executed for the first version of the system that will undergo further development to fit the specific business. The usability testing was conducted with a sample of 5 participants with the following characteristics:

- The average age of the participants was 34 years old.
- Every participant had a smartphone.





*Machines can either work on some order or wait. There are also 2 operators and one technician, who work in the production hall. The production hall has a TV monitor that displays the current state of all the machines.*

*Your boss introduced you to the new system that calculates an optimal production plan. The system shows the overview of all machines and their current state on the TV screen at the production hall. The same system is available as an application on your work tablet if you need to interact with the production overview or see the calculated production schedule.*

*Your job is to watch over production and deal with the events during the day. Your main priority is to deliver all orders correctly and meet the deadline.*

After the orientation, a participant was presented to the test environment consisted of two elements:

- **TV screen** - working as a second monitor for the computer. The screen displayed the first instance of the frontend application that was running in the browser as a web page.
- **Tablet application** - the second instance of the frontend application was running on the Android tablet as a mobile application.

The testing setup is shown in Figure 7.2. Each participant was given a set of the tasks presented in Appendix C.

The researcher observed the participant's usage of the system and took the notes as needed. After all the tasks were completed, the researcher conducted a post-test interview with the participant. Some questions used a Likert scale[50] from 1 to 5 (1 being a negative experience, 5 being a positive

experience), others an open answer. The participant's answers to the following questions were recorded and summarized in Section 7.4. The content of the post-test interview is presented in Appendix D.

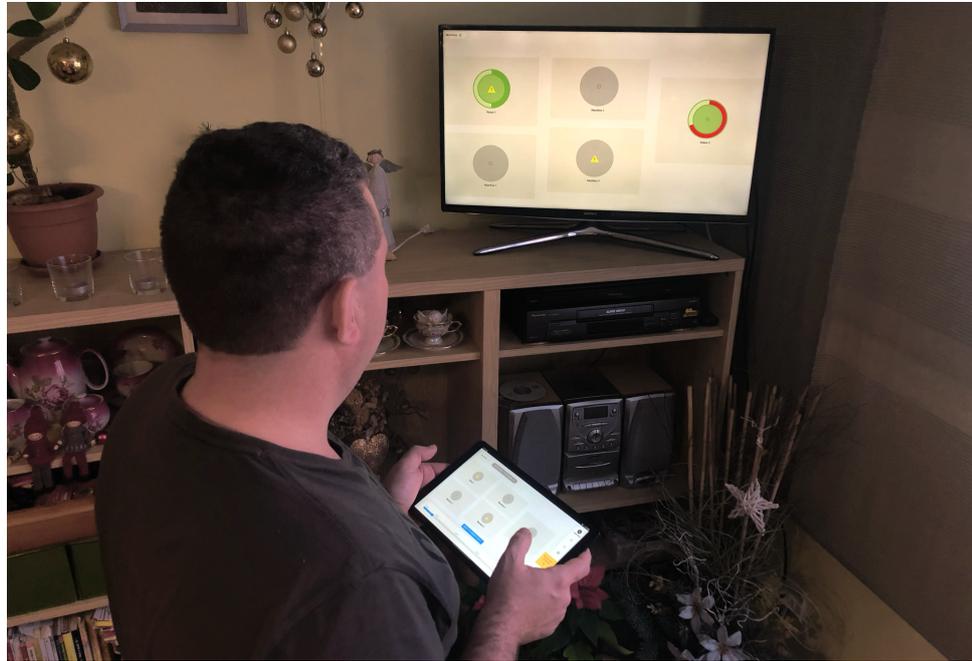


Figure 7.2: Usability testing setup

## 7.4 Results

This section describes the results of the conducted usability testing. The results are separated into three groups: identified issues and recommendations, positive findings to maintain what is working well through further development[51], and satisfaction results collected by the post-test interview.

### 7.4.1 Issues and recommendations

Although all participants completed all the tasks, the testing revealed some usability issues with the system under the test. Since no critical issues preventing the user from completing the testing scenario were found, the following list contains only issues of two severity levels:

- **serious** - the issue that was frustrating and may discourage the user from using the system,
- **minor** - the issue that some participants pointed out as a suggestion for improvement.

The list of identified issues is:

1. *Serious.* After zooming in on the machine's element in the production overview by a single tap on a tablet screen, every participant expected that the zooming out action would be performed by tapping (or double-tapping) the screen again.

**Recommendation:** Enable zooming-out the production overview by tapping and double-tapping on the tablet screen (clicking and double-clicking on the web page, respectively).

2. *Serious.* When exploring the production's past and present states, three participants confused the timeline range setting with the date selector.

**Recommendation:** Move the timeline range setting from the top area to the side of the timeline, use a clickable setting icon instead of the date range.

3. *Serious.* When exploring the past or future production state, three participants tried to use the buttons that were disabled or forgot that they moved the timeline pin and tried to work with the production overview as if it was showing its current state.

**Recommendation:** Make the production overview appear different when showing the past or future state of production. Add explanatory labels "Showing past events" or "Showing future schedule." Change the background to appear darker (or lighter for the dark color theme) than usual. Disable or remove the CTA from the machine detail when showing the past or future production state. Disable all controls except for the timeline pin, so the only option the user has is to go back to showing the current state of production.

4. *Serious.* Setting the start of the new schedule was challenging for one participant. They did not figure out that they could set the start time by clicking on it. Additionally, the date and time picker stretched the modal window out of screen bounds, so the button for going to the next step was not visible.

**Recommendation:** Add a button to change the new schedule's start time to open the date and time picker. Make the picker open in a modal window instead of a collapsible accordion.

5. *Minor.* Three participants had trouble noticing the "Reset zoom" button since it was placed in the bottom right corner, where their hand hid it most of the time.

**Recommendation:** Move the "Reset zoom" button to the top area of the screen.

6. *Minor.* Recognizing different levels of notifications took more time for two participants.

**Recommendation:** Add an icon to identify the nature of notifications (warning, breakage, check-mark icon).

7. *Minor.* After filtering the list of problems by the machine, it was not possible to return to the unfiltered list of problems.

**Recommendation:** Add a button to clear the filter.

8. *Minor.* It took some time for one participant to notice the list of problems below the filter.

**Recommendation:** Collapse a problem filter by default. Add a button to toggle the collapsed filter.

9. *Minor.* When discovering the amount of product produced the day before, two participants expected to see this information by clicking the "Order detail" button; however, the order's detail displayed the produced amount for the current date only.

**Recommendation:** Add additional information to the order's detail, which will show the order's state and produced amount valid for the focused date in the timeline.

10. *Minor.* When recalculating the schedule, one participant intended to move to the second step by clicking on the "Step 2" label.

**Recommendation:** Make the step labels clickable.

11. *Minor.* One of the participants forgot about having an interactive timeline in the system. They noticed it at first, but then it blended with the rest of the system.

**Recommendation:** Make the interactive timeline appear more noticeable by making the current time bubble bigger, adding a label on top of the timeline, or inverting the colors to achieve a pop-out effect.

#### 7.4.2 Findings

Apart from identified issues, usability testing brought out also some positive insights.

1. All the participants successfully recognized the displayed machines' states from the TV screen.
2. For one participant, the pop-out effect worked more for the warning signs in the machines' element than for the red progress bar of the delayed order. Either way, these two factors were supposed to pop out for the user, and they did.
3. All the participants managed to control the interactive timeline to see the past and future states of production.

4. All the participants managed to find the time of order delay and figure out the reason.
5. Three participants were confused that the frontend application was not fully functional (they could not add or edit information in the system). The inability to recognize the frontend application from the fully working system indicated the quality of the implemented high-fidelity prototype.
6. One participant was not used to working with tablet devices but found the system easy to use even without having previous experience with similar systems.
7. Three participants were happy to know that they could use the system as a web application on their computer.
8. Four participants described the frontend application as "visually appealing."
9. One participant pointed out the functionality to filter the problems by the machine to be the most useful in the system.
10. Three participants liked the idea of having notifications about the events at the factory, especially about the new problems assigned to them.
11. One participant who had previous experience in the manufacturing field especially appreciated the functionality of assigning a problem to another user without personally communicating with them.
12. The participants used the daily events calendar to explore the past and the future state of production as an alternative to the interactive timeline.
13. One participant said they would like to see a list of the currently executed orders for the focused date in addition to the visualized machines' states.
14. One participant pointed out that they would rather have the machines' elements arranged by the type of product they produce instead of their actual placement in the factory.
15. Two participants would like to have some extended policy for setting a problem's state to "Closed." An additional problem state "In revision" could be added to the system, which will require the creator of the problem to check if the problem was actually resolved.

### 7.4.3 Satisfaction results

Satisfaction results come from the post-test interview evaluation.

- On average, participants found the current production state very clear - 5 points on a Likert scale.

- On average, the future production plan was easy enough to figure out - 4 points on a Likert scale.
- On average, participants found managing the problems very enjoyable - 5 points on a Likert scale.
- The most enjoyable feature was the interactive timeline for four out of five participants and the zooming effect for one participant.
- The least enjoyable feature was the confusing appearance of the past and future production states, the inability to zoom out on tap, and the poor placement of the "Reset zoom" button.

#### ■ 7.4.4 Summary

The overall usability testing was a success. Although the timing of a learning process was not being precisely measured, the learning process was fast since all the participants quickly figured out where to find different kinds of information. If the participant's first attempt to perform the desired action was wrong, the second was usually right and successfully led to the task completion. Most of the first "wrong clicks" were caused by some of the issues listed above. Implementing the recommendations for these issues will prevent the first "wrong clicks." Every participant generally enjoyed using the system and found the interactive timeline fun. Participants also found the system to be useful for managing problems in production and easy to use for schedule planning purposes.



## Chapter 8

### Conclusion

This thesis aimed to help manufacturing businesses transition into the I4.0 format by offering an intuitive visualization of the calculated production schedule. The tool would encourage companies to use optimization algorithms without learning anything about it.

The ideating and development process implemented the User-Centered design technics by focusing on the end-user in every stage. The ideation phase consisted of user research conducted with people from the manufacturing industry. By contacting various businesses and interviewing representatives of the different job positions, the researcher explored the wide range of problems they encounter daily. The research revealed that an easy-to-use and accurate visualization of the production schedule is not the only goal for the people who work in the industry. Notifying the managers about the events in the production, and allowing them to deal with the daily problems, would help the production processes run smoothly and stress-free.

The concept of design presented in Chapter 4 aimed to define the necessary features to cover essential functional requirements determined by the user research. Functional requirements uncovered by the design are the subject of future work. The designed system was evaluated with a potential end-user and confirmed its clarity in displaying the algorithm's output and usefulness for the manufacturing industry workers.

After analyzing various ways to implement the high-fidelity prototype of the visualization tool, the hybrid technology was selected. It allowed the implementation of the web, desktop, and mobile applications using a single codebase. Although the primary way to use it is a tablet application, the user can access the visualization tool through the web or desktop, which can be helpful when other functionalities are added to the created tool.

Usability testing revealed four serious usability issues and seven minor. Serious usability issues were fixed by implementing the recommendations into the high-fidelity prototype. The latest version attached to this thesis is ready to be connected to other system parts, such as a backend application implemented due to the REST API specifications described in Section 6.5 and schedule optimization algorithm.

## 8.1 Future work

This section discusses the future work necessary for integrating a designed solution into some manufacturing businesses.

### 8.1.1 Implementing mocked parts of the system

This thesis focused on designing and implementing the frontend part of the complex system. That part contained visualization of the optimization algorithm's output and UI for the business processes. The essential elements necessary for making the system complete are:

- **optimization algorithm** - to calculate production schedule by the user's parameters,
- **backend application** - to allow the frontend application to use the algorithm and handle business processes.

Implementing these parts and connecting them with the frontend application will create a valuable tool for manufacturing businesses to optimize the production schedule and handle daily operations.

### 8.1.2 Analyzing the impact

Having a high-fidelity prototype of the system is not enough to tell whether it will benefit the manufacturing businesses or beat the similar existing solutions. After implementing the mocked parts, it will be possible to perform performance analysis comparing the created visualization tool to existing applications.

### 8.1.3 Extending the algorithm usage

Created frontend application focused only on working with a scheduling algorithm output. It considered the calculated schedule for a sample factory and did not focus on how that schedule was created. Other valuable functionalities of the frontend application would be:

- to allow the user to set up the algorithm conditions,
- to allow the user to choose from various calculated outputs depending on the optimization parameter (i.e., between the schedule with violated deadlines and finished orders and the schedule with met deadlines and incomplete orders),
- to handle asynchronous schedule calculation in case of a longer computation time.

#### ■ 8.1.4 Extending the system functionalities

Due to a wide range of the functional requirements determined by the user research, some desired features were omitted from the design and implementation. Future work could focus on adding these features to the system:

- machine management feature allowing to add, remove and edit the machines in the system,
- a graphic editor feature allowing to set the production overview by placing the machine's elements in their places,
- user management feature allowing to add, remove and edit the users in the system,
- user authentication feature,
- material management feature for handling the warehouse resources.

Creating a minimalistic functional solution for a specific manufacturing business would be a good first step in future work. After that solution is successfully integrated and proved beneficial, the work could be continued to make the visualization tool universal.



# Appendix A

## User Research Interview Scenario

### 1. The mental model of the production process

- a. What does the delivery process of one order look like (from the order to the delivery of the final product)?
- b. Are priorities important? If so, by what criteria something gets a higher priority than the others?
- c. How is the end of a specific part (or the whole process) of the production announced? (an authorized employee reports the end of each specific part, or is there one person who monitors the whole process and what is done and what is not?)
- d. How is a residual material, if any exists, stored?
- e. What is the definition of a successful order? (Everything was done on time, something was done faster, it took longer, but it has good quality, is it important to have many orders, and therefore money?)

### 2. The role of people in the production process

- a. How many people are involved in one production process?
- b. What does the hierarchy between these people look like?
- c. How are the responsibilities divided between these people? (Are they only interested in their dedicated part, or do they also have an overview of the production context?)
- d. Who is familiar with the scheduled schedule and how far ahead?
- e. Try to remember a case where the production process was affected by a human factor or human error (material delay, sudden illness, missed work, etc.). Does it happen often?

### 3. The role of software in the production process

- a. What tools are used to keep track of priorities and subsequently monitor their fulfillment?
- b. What tools are used to display the calculated schedule?
- c. What tools are used to display the current status of the production?

- d. How is the finished part of the work recorded? How often are these records used?

**4. Main problems during the production process**

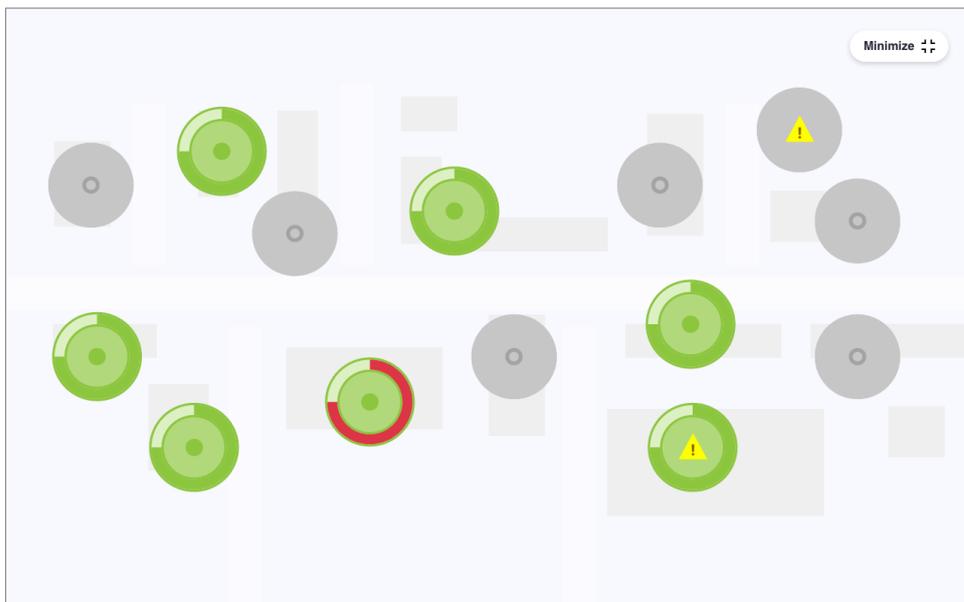
- a. What was the last time something took longer than it should?
- b. What was the last problem that was solved, and how was it solved?
- c. What is the source of these problems? (People, organization, management, a client?)
- d. What is the one thing they would like to change in their work?

**5. How the problems are solved**

- a. How was the delay in order completion last detected, and how was it addressed?
- b. Who is responsible for the order's delay (or other mentioned problems)?
- c. What caused the delay (or other problem)?
- d. Was there an effort to prevent emerging problems (e.g., optimize the process, plan the process ahead, use software solutions)?

## Appendix B

### Designed Screens



**Figure B.1:** TV view of the production overview.

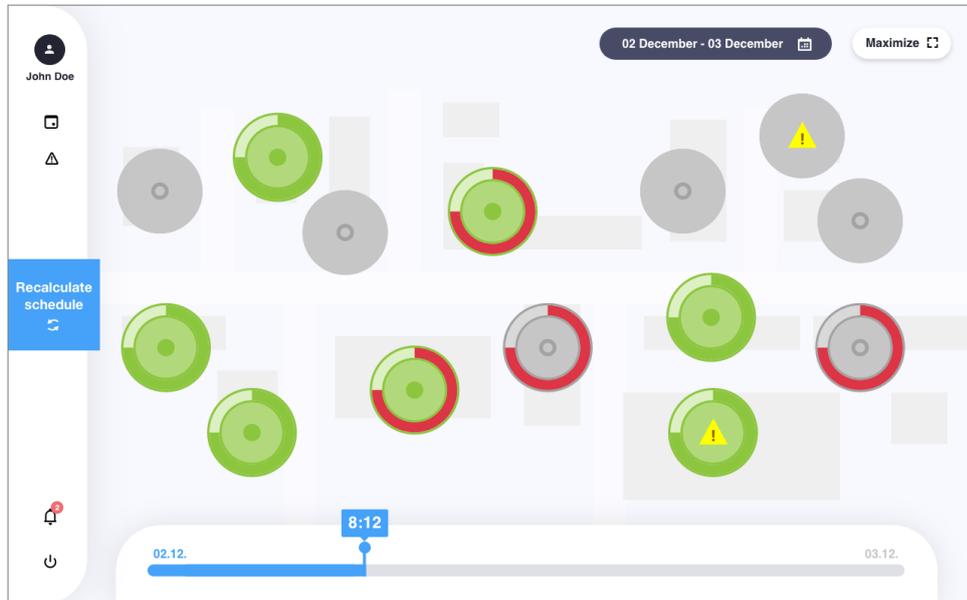


Figure B.2: App view of the production overview.

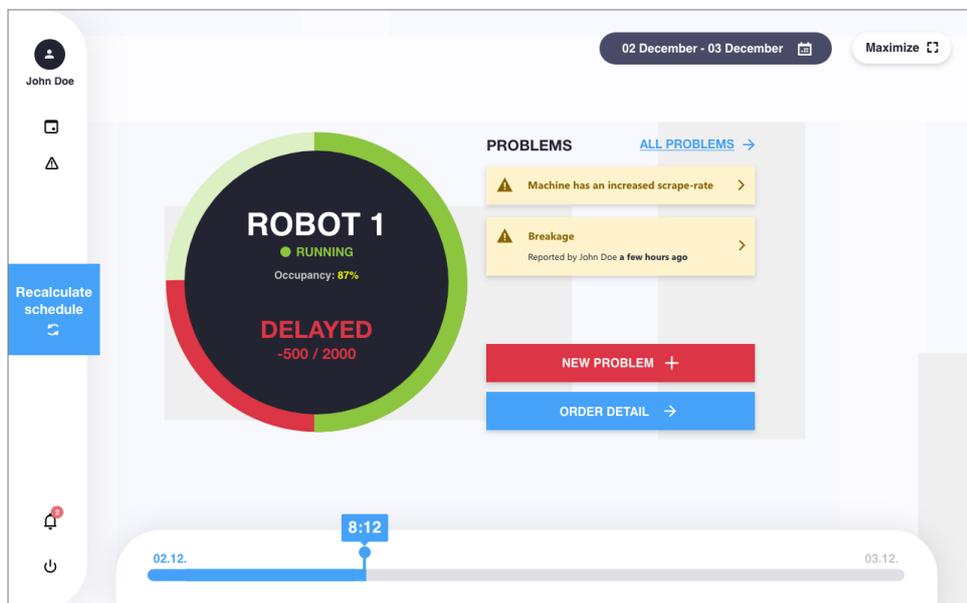


Figure B.3: Detailed machine view after zooming in production overview.

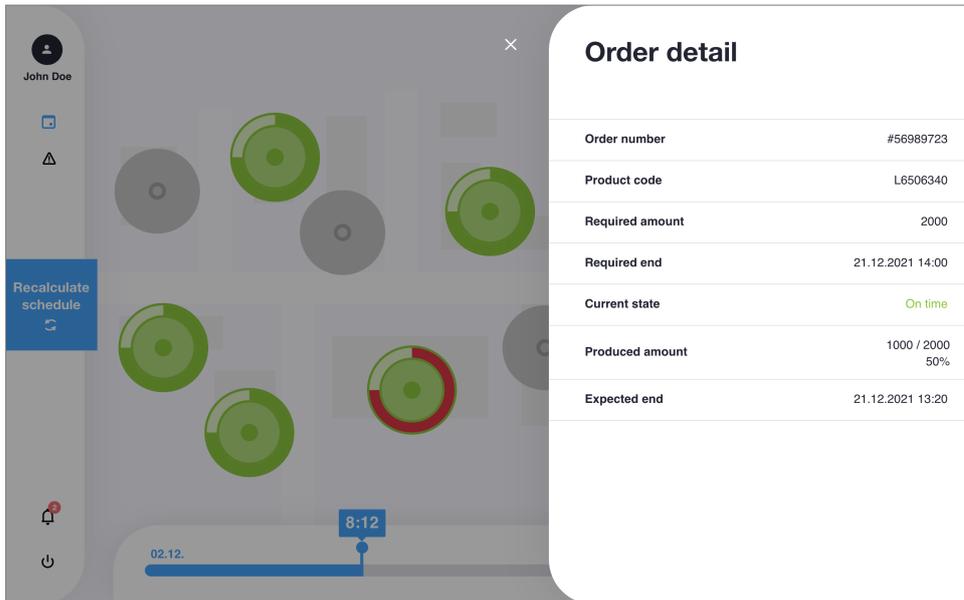


Figure B.4: Order detail view.

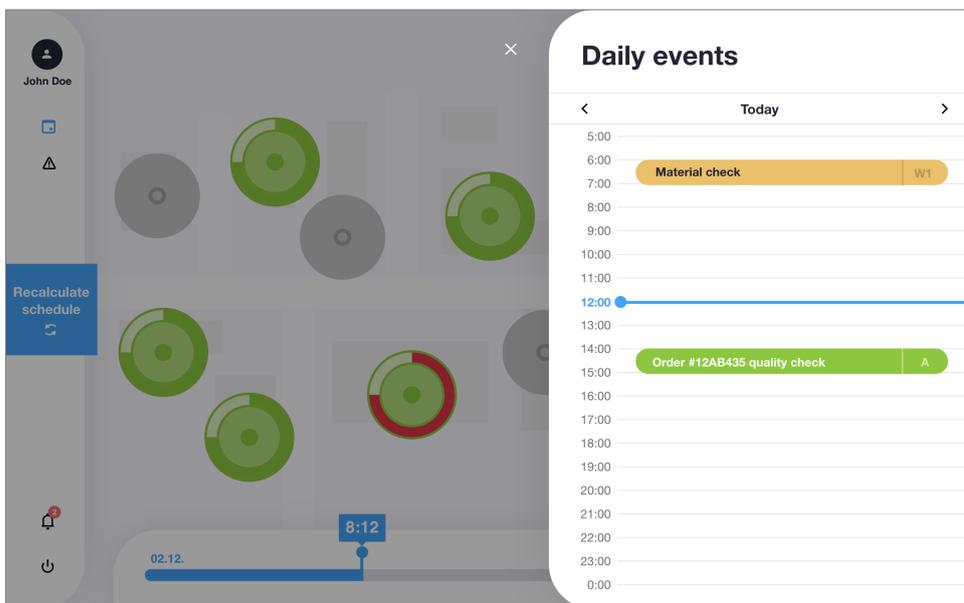


Figure B.5: Daily events view.

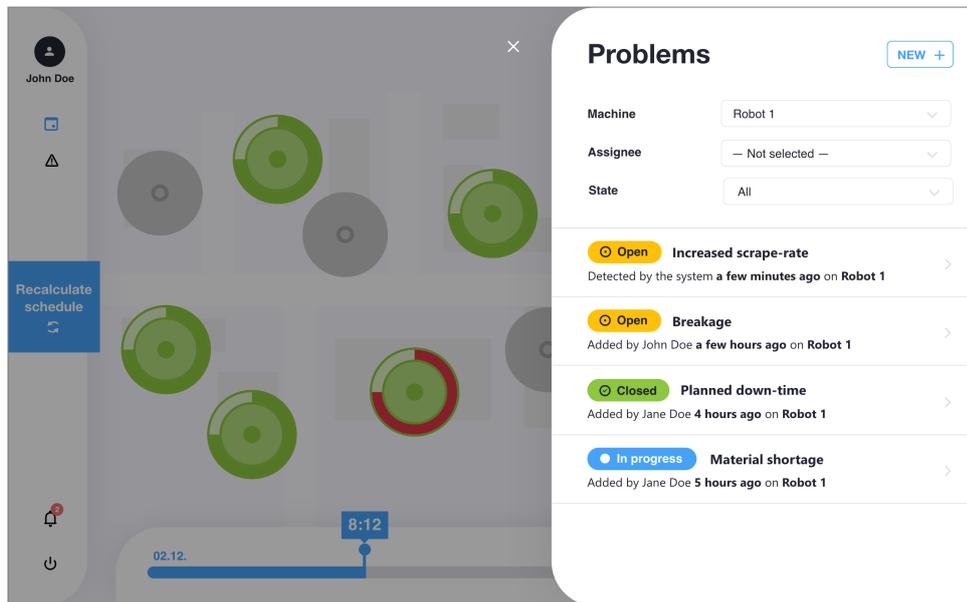


Figure B.6: Problem list view.

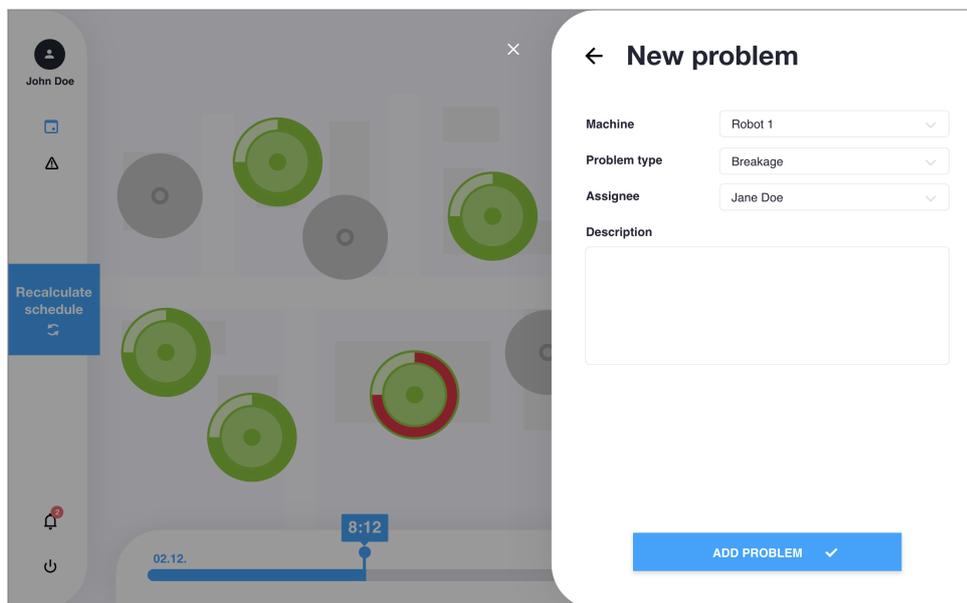


Figure B.7: New problem view.

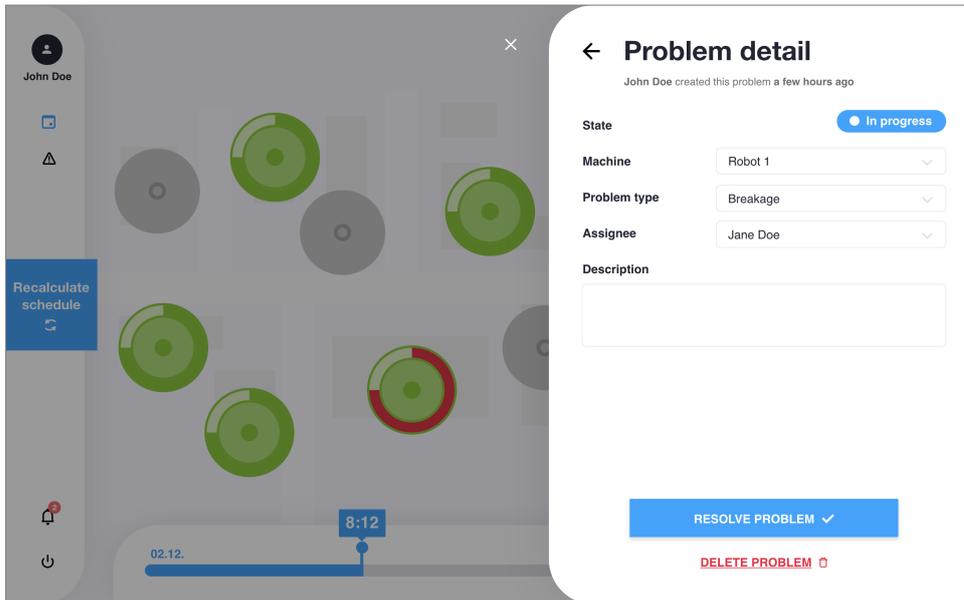


Figure B.8: Problem detail view.

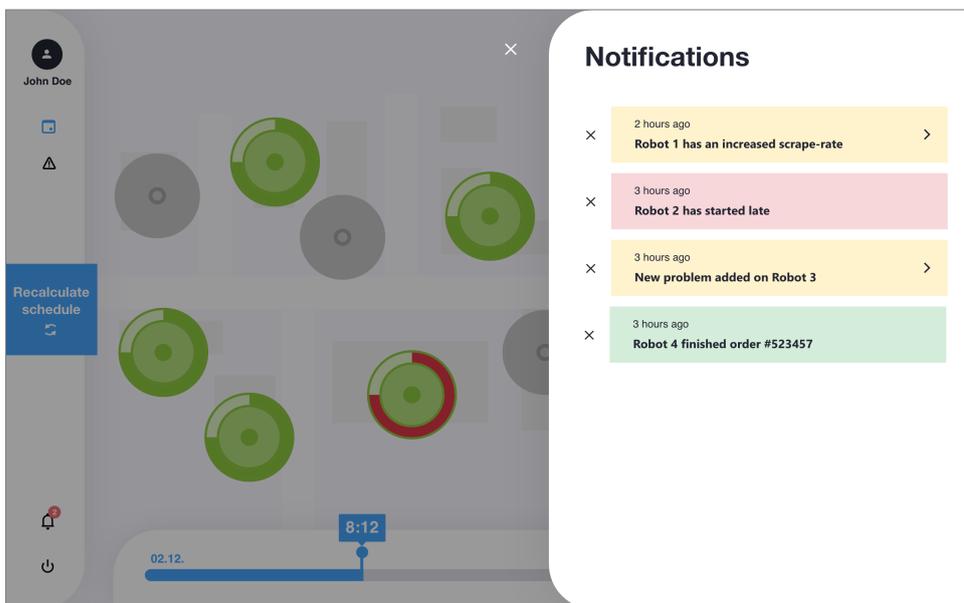


Figure B.9: Notification list view.

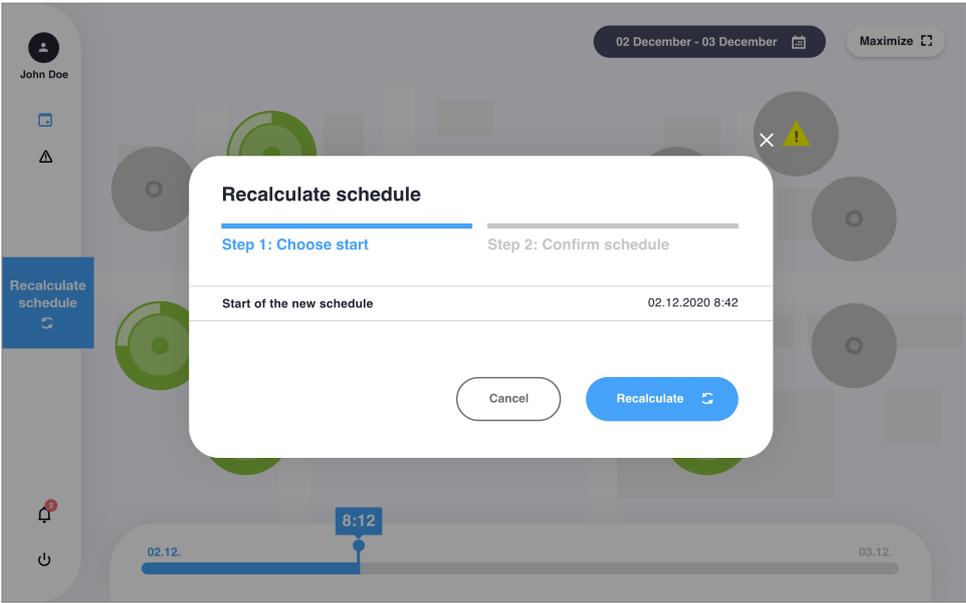


Figure B.10: Schedule recalculation dialog - step 1.

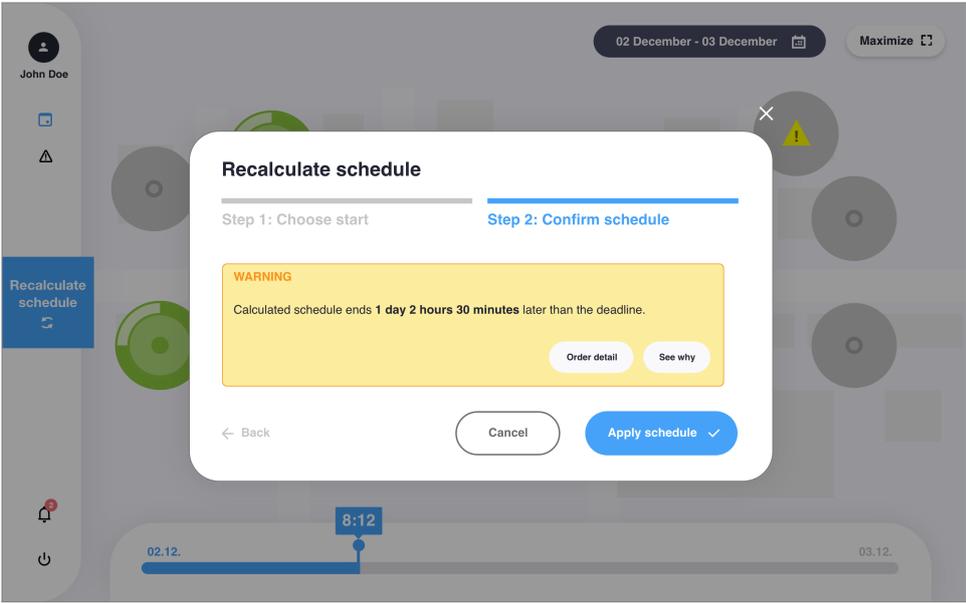


Figure B.11: Schedule recalculation dialog - step 2.



## Appendix C

### Usability Test Tasks

- Task 1:** Look at the current state of the ongoing production displayed on the TV screen. What did you notice first? Why? Can you determine which machines are currently manufacturing the product and which ones are waiting?
- Task 2:** Use the system to see what machines will be engaged in the production tomorrow at the same time. Explore how many pieces was produced on the Robot 1 yesterday at the same time.
- Task 3:** There is a problem on the Machine 3. Use the system to assign this problem to the technician.
- Task 4:** Use the system to figure out when and why delay occurred on the Robot 2.
- Task 5:** Use the system to calculate a new production schedule that will start 1 hour from now.
- Task 6:** Describe what the system notifies you about.
- Task 7:** Use the system to describe what events will occur during the day.



## Appendix D

### Post-Test Interview

Answer the following questions about your experience with the system.

1. In the system it was clear what was happening in the production hall at the present moment.

**Strongly disagree**

**Disagree**

**Neutral**

**Agree**

**Strongly agree**

2. It was easy to understand what the production state was for the past and the future moment by using the system.

**Strongly disagree**

**Disagree**

**Neutral**

**Agree**

**Strongly agree**

3. It was enjoyable to use the system for handling problems.

**Strongly disagree**

**Disagree**

**Neutral**

**Agree**

**Strongly agree**

4. What was the most enjoyable part of using the system for you?

5. What part of the system was the least enjoyable to use?



# Appendix E

## Usage Manual

### E.1 Prerequisites

1. Download and install Node.js: <https://nodejs.org/>.
2. Download and install Android Studio for Android development: <https://developer.android.com/studio>.
3. Download and install XCode for iOS development (macOS only): <https://developer.apple.com/xcode/>.
4. Install Ionic CLI globally on your machine by running this command in terminal: `npm install -g @ionic/cli`.
5. Install Angular CLI globally on your machine by running this command in terminal: `npm install -g @angular/cli`.
6. For MacOS install pod by running this command in terminal:  
`sudo gem install cocoapods`.

### E.2 Running the project

- To start a **development server**, open a terminal window in the project's root folder and run: `npm run start`. The server will be available on your local machine at <http://localhost:4200>.
- To build an **Android** application, open a terminal window in the project's root folder and run: `npm run build-android`. Then, open Android Studio in `android/` folder and run the project on an Android device or emulator.
- To build an **iOS** application, open a terminal window in the project's root folder and run: `npm run build-ios`. Then, open XCode in `ios/` folder and run the project on an iOS device or emulator.
- To build a **web** application, open a terminal window in the project's root folder and run: `npm run build-prod`. Web assets will be available in `www/` folder.





## Appendix F

### Bibliography

- [1] Flaticon, “Free Icons and Stickers - Millions of resources to download.” [Online]. Available: <https://www.flaticon.com/>
- [2] Freepik, “GFreepik: Free Vectors, Stock Photos PSD Downloads.” [Online]. Available: <https://www.freepik.com/>
- [3] W. W. Royce, *Managing the Development of Large Software Systems*. Technical Papers of Western Electronic Show and Convention, 1970.
- [4] D. Hix, *Developing User Interfaces (Wiley Professional Computing)*, 1st ed. John Wiley & Sons, 1993.
- [5] “Manufacturing execution system for capacity planning and scheduling of production, production management, maintenance, production logistics and data collection from machines | MES PHARIS.” [Online]. Available: <https://www.pharis.cz/en/about>
- [6] “WorkMonitor: Sledování efektivity výroby.” [Online]. Available: <https://papouch.com/workmonitor-sledovani-efektivita-vyroby-p4724/>
- [7] Bundesverband der Deutschen Industrie e.V., “Innovation / Leben 4.0.” [Online]. Available: <https://bdi.eu/leben-4.0/innovation/>
- [8] A. Aridi and D. Querejazu, *Manufacturing a Startup: A case study of Industry 4.0 development in the Czech Republic*, 2019.
- [9] I. Sommerville, *Software Engineering*. Pearson, 2011.
- [10] B. Gremillion, “Between ‘Wireframe’ and ‘Mockup’ Lies a World of Nomenclature,” 04 2020. [Online]. Available: <https://www.uxpin.com/studio/blog/wireframe-mockup-lofi-hifi-nomenclature/>
- [11] “The Differences Between Low Fidelity vs. High Fidelity Prototyping,” 07 2021. [Online]. Available: <https://www.protopie.io/blog/low-fidelity-vs-high-fidelity-prototyping>
- [12] F. W. T. Wallace Clark, Walter N Polakov, *The Gantt chart, a working tool of management*. New York, The Ronald Press Company, 1922.

- [13] M. Addo and W. Eboh, *Qualitative and Quantitative Research Methods*, 01 2014, p. 148.
- [14] M. Miovisky, *Kvalitativní přístup a metody v psychologickém výzkumu*. Grada Publishing, a.s., 01 2006.
- [15] M. Kuniavsky, *Observing the User Experience: A Practitioner's Guide to User Research*, ser. Interactive Technologies. Elsevier Science, 2003.
- [16] B. R. Lunsford, Thomas R.; Lunsford, "The Research Sample, Part I: Sampling," *JPO Journal of Prosthetics and Orthotics*, vol. 7, no. 3, 1995.
- [17] "Průmysl a výroba - Firma, velkoobchod, výroba - ČR - Atlasfirem.info," 2021. [Online]. Available: <https://www.atlasfirem.info/katalog/firma-velkoobchod-vyroba/prumysl-a-vyroba/stat-1-ceska-republika-strana-1.html>
- [18] C. Lewko and J. Parton, "Developer Personas," *Developer Relations*, pp. 131–139, 2021.
- [19] A. Cooper, *About Face: The Essentials of Interaction Design*. John Wiley Sons, Inc., Indianapolis, Indiana, 2014.
- [20] "How to Write Problem Scenarios | Solution Tree Blog," 02 2016. [Online]. Available: <https://www.solutiontree.com/blog/how-to-write-problem-scenarios/>
- [21] N. N. Group, "Storyboards Help Visualize UX Ideas," 2021. [Online]. Available: <https://www.nngroup.com/articles/storyboards-visualize-ideas/>
- [22] C. L. Prototypes, "Storyboard That: The World's Best Free Online Storyboard Creator," 2021. [Online]. Available: <https://www.storyboardthat.com/>
- [23] "What are User Scenarios?" 2021. [Online]. Available: <https://www.interaction-design.org/literature/topics/user-scenarios>
- [24] "User flow," 2021. [Online]. Available: <https://www.optimizely.com/optimization-glossary/user-flow/>
- [25] "Adobe XD | Fast Powerful UI/UX Design Collaboration Tool," 2021. [Online]. Available: <https://www.adobe.com/cz/products/xd.html>
- [26] "Tablet Operating System Market Share Worldwide | Statcounter Global Stats," 2021. [Online]. Available: <https://gs.statcounter.com/os-market-share/tablet/worldwide>
- [27] "Progressive web apps (PWAs) | MDN," 10 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps)

- [28] Aureate Labs, “The Crucial limitations of PWA in IOS,” 04 2021. [Online]. Available: <https://yourstory.com/mystory/crucial-limitations-pwa-ios/amp>
- [29] R. Enihe and J. Joshua, “Hybrid mobile application development: A better alternative to native,” 05 2020.
- [30] “React Native · Learn once, write anywhere,” 2021. [Online]. Available: <https://reactnative.dev/>
- [31] “HTML: HyperText Markup Language | MDN,” 10 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [32] “CSS: Cascading Style Sheets | MDN,” 09 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [33] “Introduction to the DOM - Web APIs | MDN,” 09 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
- [34] “Flutter - Build apps for any screen,” 2021. [Online]. Available: <https://flutter.dev/>
- [35] Ionic, “Cross-Platform Mobile App Development,” 2021. [Online]. Available: <https://ionicframework.com/>
- [36] A. Cordova, “Mobile apps with HTML, CSS JS. Target multiple platforms with one code base. Free and open source.” 2021. [Online]. Available: <https://cordova.apache.org/>
- [37] Capacitor, “Open source native runtime for building Web Native apps.” 2021. [Online]. Available: <https://capacitorjs.com/>
- [38] “Angular - The modern web developer’s platform,” 2021. [Online]. Available: <https://angular.io/>
- [39] “JavaScript With Syntax For Types.” 2021. [Online]. Available: <https://www.typescriptlang.org/>
- [40] “React – A JavaScript library for building user interfaces,” 2021. [Online]. Available: <https://reactjs.org/>
- [41] “Vue.js,” 2021. [Online]. Available: <https://vuejs.org/>
- [42] “The WebSocket API (WebSockets) - Web APIs | MDN,” 09 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)
- [43] “Typescript Declaration Merging,” 2021. [Online]. Available: <https://www.typescriptlang.org/docs/handbook/declaration-merging.html>

- [44] T. Willison, “Panzoom - A library for panning and zooming elements using CSS transforms.” 2021. [Online]. Available: <https://timmywil.com/panzoom/>
- [45] GitHub, “Websites for you and your projects.” 12 2021. [Online]. Available: <https://pages.github.com/>
- [46] D. Garlan and M. Shaw, “An Introduction to Software Architecture,” Tech. Rep., 1994. [Online]. Available: [https://www.cs.cmu.edu/afs/cs/project/able/ftp/intro\\_softarch/intro\\_softarch.pdf](https://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf)
- [47] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, 2000. [Online]. Available: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [48] T. Berners-Lee, “Uniform resource identifier (URI): Generic syntax.” 01 2005. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc3986>
- [49] *A mathematical model of the finding of usability problems*. New York, New York, USA: ACM Press, 1993.
- [50] S. C. Allen, I. Elaine, “Statistics Roundtable: Likert Scales And Data Analyses,” *Quality Progress*, vol. 40, pp. 64–65, 2007.
- [51] Assistant Secretary for Public Affairs, “Reporting Usability Test Results | Usability.gov,” 2021. [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/reporting-usability-test-results.html>