

**CZECH TECHNICAL UNIVERSITY IN PRAGUE**  
**FACULTY OF MECHANICAL ENGINEERING**

**Department of Enterprise Management and Economics**



**MASTER'S THESIS**

**Use of 2D Codes in Enterprise Processes**

**Využití 2D kódů v podnikových procesech**



# MASTER'S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Vencl Štěpán** Personal ID number: **459660**  
Faculty / Institute: **Faculty of Mechanical Engineering**  
Department / Institute: **Department of Management and Economics**  
Study program: **Mechanical Engineering**  
Branch of study: **Enterprise Management and Economics**

## II. Master's thesis details

Master's thesis title in English:

**Use of 2D Codes in Enterprise Processes**

Master's thesis title in Czech:

**Využití 2D kódů v podnikových procesech**

Guidelines:

1. Introduction – motivation, goals and objectives for the thesis
2. Theoretical part - process management, enterprise processes, 1D and 2D codes, their functionality and use, alternative technologies, Python introduction
3. Practical part - definition of the problem and limiting conditions, selected enterprise process mapping, Python application for code reading
4. Conclusions and recommendations

Bibliography / sources:

1. VOM BROCKE, Jan. Handbook on Business Process Management 1: Introduction, Methods, and Information Systems. 2nd ed. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2014. ISBN 978-3-642-45099-0.
2. SHARP, Alec a Patrick McDERMOTT. Workflow modeling: tools for process improvement and applications development. 2nd ed. Boston: Artech House, c2009. ISBN 978-1-59693-192-3.
3. International Organization for Standardization (2015). Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification (ISO/IEC 18004:2015(E)).
4. KATO, Hiroko. Barcodes for mobile devices. Cambridge: Cambridge University Press, 2010. viii, 257 s., [19] s. příl. ISBN 978-0-521-88839-4.

Name and workplace of master's thesis supervisor:

**doc. Ing. Jan Horejc, Ph.D. Department of Management and Economics FME**

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

**Ing. Pavel Scholz Department of Management and Economics FME**

Date of master's thesis assignment: **31.03.2022** Deadline for master's thesis submission: **22.07.2022**

Assignment valid until: **29.09.2023**

\_\_\_\_\_  
doc. Ing. Jan Horejc, Ph.D.  
Supervisor's signature

\_\_\_\_\_  
Ing. Miroslav Žilka, Ph.D.  
Head of department's signature

\_\_\_\_\_  
prof. Ing. Michael Valášek, DrSc.  
Dean's signature

## III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his 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

## **DECLARATION**

I declare that this thesis was independently composed by myself,  
exclusively using the sources and literature listed in the Bibliography.

In .....

Bc. Štěpán Vencl

## **ACKNOWLEDGEMENTS**

I would like to thank my supervisor Pavel Scholz from the Czech Technical University in Prague (CTU) for his guidance and helpful advice throughout this project. To Inga Björk Guðmundsdóttir from the Reykjavík University (RU) for her willingness to help a student in a foreign country. I would also like to thank everyone from academia who is dedicated to the personal growth of students. To my alma mater for allowing me to study abroad at RU.

I thank my parents for nurturing my intellectual pursuits. To all my close ones for the bonds we have, and for the patience they have with me.



## **ABSTRACT**

This master's thesis presents business processes and the ways in which they can be modeled and managed. Different types of optical codes can be used to increase the efficiency of such processes. These codes are described in detail and compared with alternative technologies. At the APS Systems company, an inefficient process is targeted, and an improved version of this process is proposed. A desktop application in the Python programming language is developed to support this process. The application is designed for components ordering and subsequent registration of delivered packages in the warehouse using specially designed QR codes. The functionality of the application is demonstrated through three videos. At the end of the thesis, the advantages of this solution are presented, and other possible improvements are discussed.

## **KEY WORDS**

Business Process Modeling, Barcodes, 2D codes, QR code, Python

<b>University:</b>	Czech Technical University in Prague
<b>Faculty:</b>	Faculty of Mechanical Engineering
<b>Field of study:</b>	Enterprise Management and Economics
<b>Thesis title:</b>	Use of 2D Codes in Enterprise Processes
<b>Author:</b>	Bc. Štěpán Vencl
<b>CTU supervisor 1:</b>	doc. Ing. Jan Horejc, Ph.D.
<b>CTU supervisor 2:</b>	Ing. Pavel Scholz
<b>External supervisor:</b>	Inga Björk Guðmundsdóttir, MSc

## **ANOTACE**

Tato diplomová práce představuje podnikové procesy a způsoby, jak je modelovat a řídit. Ke zvýšení účinnosti takových procesů lze použít různé typy optických kódů. Tyto kódy jsou podrobně popsány a porovnány s alternativními technologiemi. Ve společnosti APS Systems je zaměřen neefektivní proces a je navržena vylepšená verze tohoto procesu. Pro podporu tohoto procesu je vyvinuta desktopová aplikace v programovacím jazyce Python. Aplikace je určena pro objednávání komponent a následnou evidenci dodaných balíků ve skladě pomocí speciálně navržených QR kódů. Funkčnost aplikace je demonstrována prostřednictvím tří videí. V závěru práce jsou uvedeny výhody tohoto řešení a diskutována další možná vylepšení.

## **KLÍČOVÁ SLOVA**

Modelování podnikových procesů, čárové kódy, 2D kódy, QR kód, Python

Univerzita:	České vysoké učení technické v Praze
Fakulta:	Fakulta strojní
Studijní obor:	Řízení a ekonomika podniku
Diplomové práce:	Využití 2D kódů v podnikových procesech
Autor:	Bc. Štěpán Vencl
ČVUT vedoucí 1:	doc. Ing. Jan Horejc, Ph.D.
ČVUT vedoucí 2:	Ing. Pavel Scholz
Externí vedoucí:	Inga Björk Guðmundsdóttir, MSc

# Table of Contents

Introduction.....	10
How to Read This Thesis .....	11
1 Processes .....	12
1.1 Process Management.....	12
1.1.1 Evolution of BPM .....	13
1.1.2 Process Enablers .....	15
1.1.3 Advantages of BPM.....	16
1.2 Enterprise Processes .....	17
1.3 BP Modeling.....	18
1.3.1 ARIS.....	19
1.3.2 UML .....	21
1.3.3 BPMN .....	22
2 Barcodes.....	28
2.1 Barcode Evolution .....	29
2.2 Barcode Anatomy .....	31
2.3 Barcode Readers.....	32
2.3.1 Laser Scanners .....	32
2.3.2 Image-Based Readers.....	32
2.4 Symbology of 1D Codes.....	34
2.4.1 Code 39 .....	35
2.4.2 UPC / EAN / IAN .....	36
2.4.3 Code 128.....	37
3 Two-Dimensional Codes .....	39
3.1 Error Correction .....	39
3.2 Stacked 2D codes .....	40
3.2.1 PDF 417 .....	40
3.3 Matrix 2D Codes .....	41
3.3.1 MaxiCode .....	41
3.3.2 Aztec Code .....	42
3.3.3 HCCB .....	45
3.3.4 Data Matrix.....	45
3.3.5 QR Code .....	47
3.4 2D Codes for AR .....	49
3.4.1 ARToolKit.....	50

3.4.2	ARTag .....	50
3.4.3	ArUco.....	50
3.5	Utilization of 2D Codes.....	51
3.5.1	Payments & Authentication .....	51
3.5.2	Marketing & Advertising .....	52
3.5.3	Manufacturing & Logistics.....	53
<b>4</b>	<b>Alternative Technologies .....</b>	<b>54</b>
4.1	OCR.....	54
4.2	Magnetic Stripes .....	54
4.3	IC Chips .....	55
4.4	Radio Frequency Technologies .....	55
4.4.1	RFID.....	55
4.4.2	NFC.....	56
4.5	Comparison .....	56
<b>5</b>	<b>Used SW Tools .....</b>	<b>58</b>
5.1	Python .....	58
5.1.1	Python Enhancement Proposals .....	58
5.1.2	Syntax.....	59
5.1.3	Libraries.....	60
5.2	PyCharm IDE .....	60
5.3	GitHub.....	61
5.4	Lucidchart.....	61
<b>6</b>	<b>APS Systems.....</b>	<b>62</b>
6.1	Company Introduction .....	63
6.2	Organizational Structure .....	64
6.3	Products.....	64
6.4	Customers.....	66
6.5	Suppliers .....	66
6.6	Production Process .....	67
<b>7</b>	<b>Logistics &amp; Inventory Problem .....</b>	<b>68</b>
7.1	Current Situation.....	68
7.2	Inventory Flow .....	70
7.3	Motivation .....	71
<b>8</b>	<b>Logistics &amp; Inventory Solution.....</b>	<b>72</b>
8.1	Solution Proposal.....	73

8.2	New Process Workflow.....	74
8.3	Python Application.....	77
8.3.1	Creating New Order.....	78
8.3.2	Reading QR Codes .....	80
8.3.3	Support Functionalities .....	84
8.4	Code overview .....	85
8.4.1	Project structure.....	86
8.4.2	GUI.....	87
8.4.3	Functions.....	88
<b>9</b>	<b>Further Improvements .....</b>	<b>90</b>
9.1	App Upgrades .....	90
9.2	AR Demo .....	91
9.3	UI and UX Designs .....	93
	<b>Conclusion .....</b>	<b>95</b>
<b>A</b>	<b>Abbreviations &amp; Acronyms.....</b>	<b>96</b>
<b>B</b>	<b>Bibliography &amp; References.....</b>	<b>98</b>
<b>C</b>	<b>Lists .....</b>	<b>102</b>
<b>D</b>	<b>Attachments .....</b>	<b>105</b>

# Introduction

We come across them everywhere in our cultural environment. The stripiness of their predecessors has eventually evolved to a second dimension to form chessboard-looking patterns. We know what they are called. Many know how to use them. But how do they work? What are their limitations, benefits, or unused potentials?

Not only the manufacturing companies are struggling with process efficiency, we all are. And we can experience these flaws in processes in our daily lives. And to no surprise, they become more apparent when we are waiting, or even, when our time is being wasted. We ask ourselves: Why do the state administration office workers need to stamp every paper document multiple times when we live in the so-called "digital age"? How is the package I ordered last week still not here? Aren't there more efficient and faster ways to do things?

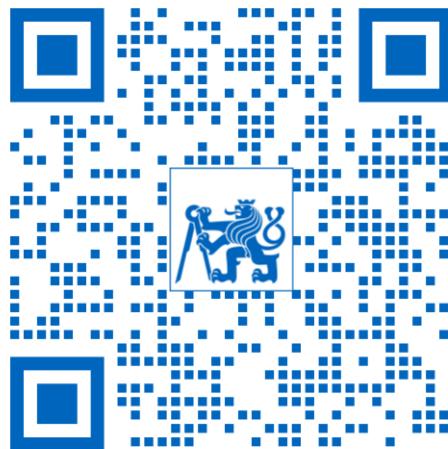
There are! One of the possible solutions is to use the technology of optical codes. These carry data that can be accessed by the reading technologies in fractions of a second. The most widespread two-dimensional optical code - Quick Response (QR) Code is named after this ability. Various types of barcodes are used for decades to help with process efficiency. We can see them everywhere. On product labels, packages, or marketing posters.

In this thesis, I am elucidating the principle of optical barcodes and describing usage for various purposes in business areas and industrial processes. My objective is to educate the reader on the topics of business process modelling, and the functionality of various barcodes.

With this acquired knowledge I aim to target selected inefficient enterprise process, design and model improved substitution, and develop software tool supporting this new process by utilizing two-dimensional codes.

# How to Read This Thesis

Throughout this thesis, these blue CTU stylized codes will occur with external links connected to my work. In electronic version of this thesis, these links can be accessed by highlighted QR# text. This first: [QR1](#) link serves as an example of this visual distinction compared to common Black and White (B&W) barcodes introduced in respective chapters.



QR 1: External link example [1]

Abbreviations are listed in section A at the end of this thesis. The First occurrence of each abbreviation is stated with the full definition. Abbreviations are typed in capital letters and linked to their definition (in the electronic version of the document). When plural is used, lowercase "-s" is appended to the abbreviation.

The electronic version of this thesis includes links to the Table of Contents (ToC) and other sections for easier navigation in the footer of each page.

The majority of included Figures are created or modified by me. If it is solely my work, no citation nor comment is included in the caption. If it is inspired by some source, "Based on [reference]" is stated in the caption.

Few Chapters include utilization examples I encountered in daily life Here in Iceland, where I was (am) writing this thesis in the final year of my studies. These are framed in stylized brackets shown in Figure 1.



Figure 1: Iceland Example brackets. Based on [2]

# 1 Processes

The processes are a vital part of every enterprise or company. In a highly competitive environment, it is the company's ability to effectively design, use and manage processes that provide a competitive advantage over others. But what are these processes and how can we manage them?

During my studies, I have been taught that: "Manufacturing or production process is a conversion of inputs into outputs under the influence of production factors." [3]

Cambridge dictionary offers more general definition: "Process is a series of actions that you take in order to achieve a result." [4]

Sharp and McDermott conclude their definition from other literature as: "A process is a collection of activities (or steps or tasks or whatever) that is a way to get something done." [5]

According to the International Organization for Standardization (ISO) standard (EN ISO 9001), a process is defined as a "Set of interacting activities that turn inputs into outputs." [6]

No matter the exact definition, processes include inputs, activity, and some outputs possibly (and hopefully) with added value. This thesis concerns Enterprise Processes (EP) or Business Processes (BP). Both terms are semantically interchangeable. But BP is being more commonly used in literature and online sources. The next chapters will discuss how we can control processes and introduces various types of these processes.

## 1.1 Process Management

To ensure the suitability, viability, and efficiency of any processes, we use a discipline called Business Process Management (BPM) that uses knowledge, skills, methods, tools, and different systems. Then we can use this "package" to identify, describe, measure, manage, evaluate, and improve processes in order to meet our own or customer needs. When business processes play a key role in managing the organization, then the organization is considered process-oriented, according to Řepa. [7][8]

### 1.1.1 Evolution of BPM

The industrial revolution completely changed the world of manufacturing. Before the invention of the steam engine, highly qualified craftworkers would have direct contact with the customer and thus could customize the hand-made products. Although the product outputs were very limited as craftsmen were reliant on their handwork and the help of apprentices. [5]

In the 18<sup>th</sup> century, Adam Smith described processes in manufactories that used labour division into small tasks to increase productivity. Manufactories like this would have high outputs but no room for customizability. Goods would become more available thanks to economy of scale. [9]

In the early 20<sup>th</sup> century, Frederick Winslow Taylor published "The Principles of Scientific Management". His work promotes standardization in order to improve efficiency. Around the same time, Henry Ford introduced moving assembly belts into plants producing "Model T" cars, in 1913. Three years later, the cost of the car is less than half of what it cost in 1908.

Walter A. Shewhart pioneered the use of Statistical Process Control (SPC) in the 1920s. Building upon Shewhart's SPC work, incorporated into managerial principles, W. Edwards Deming trains hundreds of engineers and businessmen in SPC and concepts of quality in after-war Japan. Twenty years later in the 1970s, Japan is considered the world leader in quality. This period uses the mantra of the very well-known PDCA (Plan-Do-Check-Act), also known as the Demings cycle (shown in Figure 2) is an iterative method used in business for the control and continuous improvement of processes and products. Deming and his quality principles laid the foundation for further quality and management theory and methods, such as Six Sigma, Lean, or TQM (Total Quality Management). [10][11][12][13]

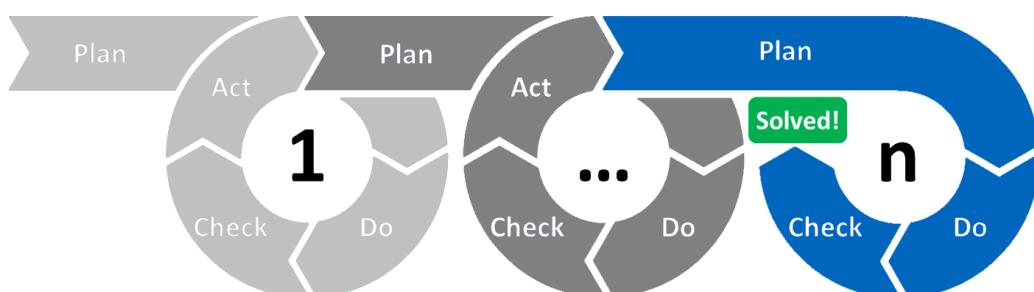


Figure 2: PDCA iterative cycle. [13]

Increasing profession specialization crystallized into the functionally oriented organizations that dominated most of the last century. Although grouping people into specialized teams can enable constant improvement of their specific strengths, this improvement takes part only in one division or department and not as a whole. These organizations, built on hierachic vertical structures can also suffer from an insufficient delegation of responsibilities or with the phenomenon known as the "Silo Mentality" depicted in Figure 3, which is a reluctance to share information or knowledge between employees or across different departments within a company, that might lead to low morale within the employees, adversely affect workflows, or even negatively impact customer experience. [6][14][15][16]

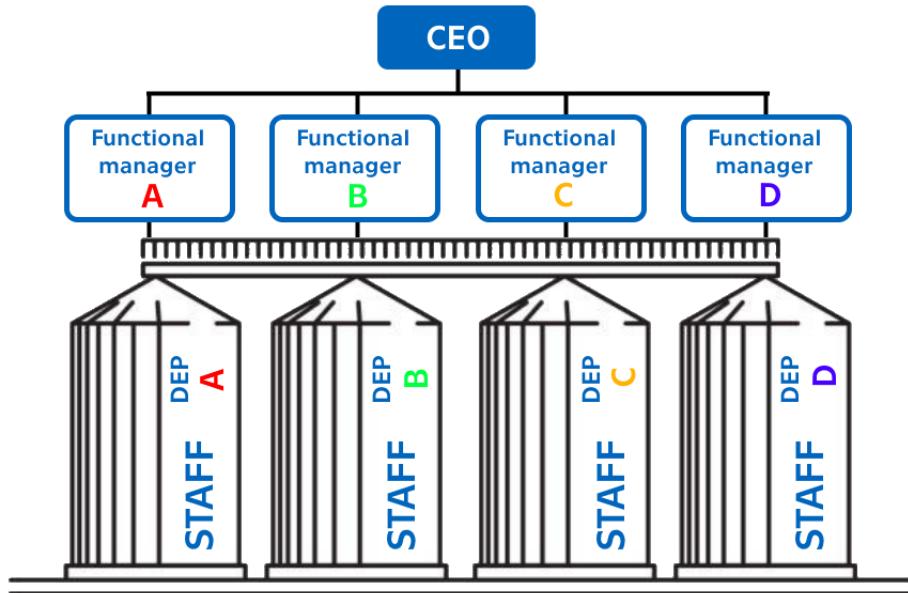


Figure 3: Silo Mentality and vertical org. structure. Based on [15].

As a reaction to the process being lost within function-organization, Michael Hammer introduces Business Process Reengineering (BPR) in the 1990s. This approach emphasizes radical transformation over the iterative, continuous improvement of processes. Hammer calls it "Undoing the Industrial Revolution". Disrupting its overspecialization enables reunifying tasks into coherent, visible processes. [5][17]

Other authors make a clear division between Improvement, which is rather small, incremental change build upon an already existing process less difficult to implement with a Bottom-up approach, and Innovation, which is radical, cultural, and structural one-time change approached Top-down. These kinds of radical changes were enabled only thanks to the advances in

technology. More performative Hardware (HW) managed to run more complex Software (SW) and these technologies became broadly available to companies and individuals. Software packages such as ERPs (Enterprise Resource Planning) and others would widespread. [7][10][18]

BPR became obsolete thanks to the onset of the information age but also because it was not very universal and sometimes implemented without common sense. As a compromise, contemporary BPM is trying to take the best from its past and merge it with available technologies to improve the enterprise's competitiveness. [7][10][18]

### 1.1.2 Process Enablers

According to Hammer, there are five critical enablers, without them, a process would be unable to operate sustainably: [12][19]

- **Process design:** The design is the specification and is the most fundamental aspect of a process. It identifies what, when, and where is to be performed by whom. Specification clarifies other enablers and their role within the process, circumstances, and information needed.
- **Process metrics:** The enterprise needs to re-evaluate if the performance metrics are not functional. Processes need end-to-end metrics that are derived from enterprise goals and customer needs. The alignment of measurable targets within metrics is a necessity.
- **Process performers:** People performing processes need a broader set of skills and behaviours compared to those working in conventional functions. Understanding the overall process and its goals, and teamwork is the key to realizing the potential of end-to-end work.
- **Process infrastructure:** IT and HR systems need to support participants of processes to effectively discharge process responsibilities. All supportive systems need to be integrated and cross-organizational.
- **Process owner:** For an end-to-end process, responsibility plays a key role, therefore processes must have their owners: foremen, team leaders, or managers with authority and responsibility for a process across the organization as a whole.

Another approach, brought by Sharp and McDermott, fit process enablers within the context of an organization as shown in Figure 4. These six are in content similar to Hammer's five critical enablers: [5]

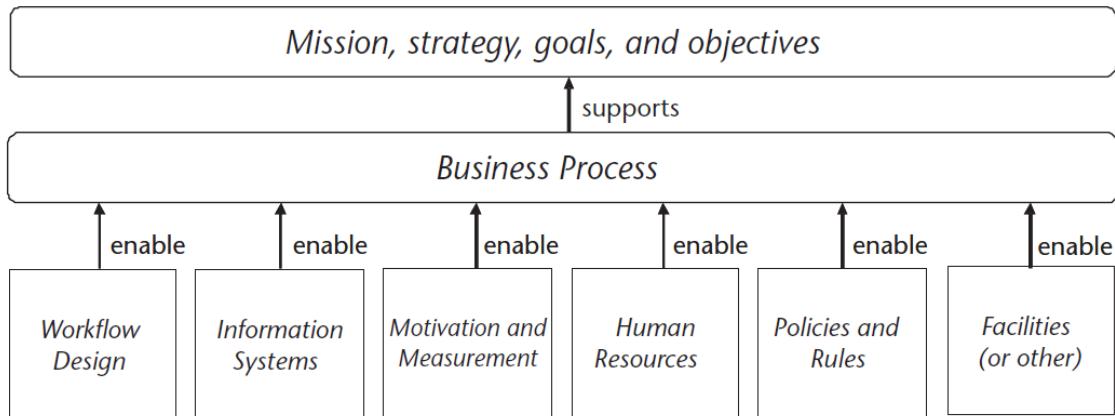


Figure 4: Sharp & McDermott's Process enablers [5]

With a wide spread of information technology in today's world, enterprise IT can no longer be just a support service but must be implemented as an integral element in the company's strategy. To support BPM, there are Business Process Management Systems (BPMS) - SW tools that apply a process-oriented approach. BPMS used to be in opposition to earlier mentioned ERP, which is more function-oriented. Once rivals in Business Intelligence (BI), nowadays modular architecture of ERP systems usually includes BPMS module within one package. [12][20]

### 1.1.3 Advantages of BPM

A process-driven approach embraced in organizations offers several benefits when compared to the traditional, functional approach. However, this does not mean, that the process-oriented approach is superior in every aspect to the function-oriented one. [5][15][16]

- Improvement and efficiency are achieved by focusing on outcomes, customer needs, and final products.
- Focus on continual improvement of process execution and process outcomes contributes to lower costs and shorter cycle times.
- Aligning and integrating processes and organizational functions leads to easier achieving planned goals and objectives.
- High overall efficiency through streamlined (and well-defined) workflow and elimination of unnecessary, redundant, or counterproductive work.

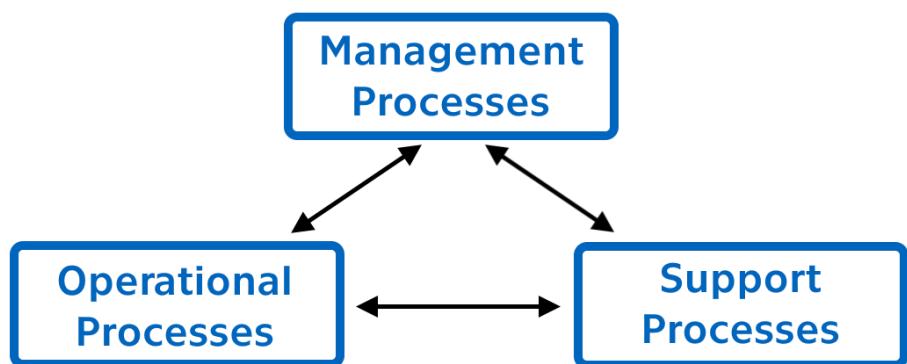
- Easy optimization, thanks to greater awareness of the whole. This also contributes to a healthier organizational ecosystem and opportunities for employee development.
- Clearly defined responsibility for each step minimizes spaces for misunderstanding and conflict.
- Empowering transparency eases new connections and relations. This can be useful for both, internal and external stakeholders.
- Related Know-How is a valuable asset and can be stored within the process, not only in the heads of employees.
- Less specialization demands higher qualifications but eliminates routine and jobs tend to be more satisfying and fulfilling.
- Defined, measurable and controlled processes can be compared against industry benchmarks and can be standardized. It also makes it easier to respond to changes.
- Overview of all activities and processes of an enterprise. Enables troubleshooting and simulation.

## 1.2 Enterprise Processes

Every enterprise or company is built on processes. Sometimes it can be hard to identify the process or even notice it, let alone describe it. As I mentioned earlier, the terms EP and BP are interchangeable. We can classify processes into 3 major groups. The names of these groups can differ in different sources, but the function and meaning are the same: [16][21][22]

- 1) **Operational Processes:** often called Main/Core/Primary, are processes that participate in creating value and income for the company. They are usually directly related to the vision and mission of the enterprise. To this group belong processes such as:
  - a) Design & Development
  - b) Manufacturing & Production
  - c) Marketing
  - d) Customer service & Support
- 2) **Support Processes:** do not participate in value generation directly, but they enable it. These processes make it possible to perform other processes effectively by creating a suitable environment and are also of strategic importance. They are usually "invisible" to an external observer. Examples might be:

- a) Accounting
  - b) IT Services
  - c) Human Resources
- 3) **Management Processes:** The previous two groups require planning, organization, coordination, monitoring, and control. Management processes also include measuring overall results and dealing with opportunities and threats from external and internal environments. They include helping and motivating employees to achieve their targets and goals.



*Figure 5: Process Types. Based on [21]*

All 3 groups depicted in Figure 5 are interconnected and integrated. We could say that without one, the other would not be able to perform well enough to survive in a business competitive environment. This simple division can be helpful for the company to identify and classify its processes.

### 1.3 BP Modeling

Business process modeling is a crucial activity of BPM. To create a suitable process for our company, we should use workflow design or simply workflow. Sharp and McDermott describe workflow as the sequence of steps, decisions, and transfers carried out by the process's participants. [5]

Whether creating an entirely new process (To-Be) or improving an already existing and functioning one (As-Is), the modeling frameworks implies following these steps: [5][20]

- 1) **Identify** (As-Is): Uncover all relations, consumed sources, and participants. Formulate purpose and consequences.
- 2) **Document** (As-Is): Graphically model workflow, parametrize input and output factors, and record all data.

- 3) **Analyze** (As-Is): Observe the impact of factors on performance. Conclude strengths and weaknesses.
- 4) **Design** (To-Be): Establish goals, scope, context, and boundaries. Model and simulate the new process, then compare it to the old one.
- 5) **Implement** (To-Be): Train participants and deploy the process to the organization. Assess suitable performance metrics.
- 6) **Control** (To-Be): Measure, analyze and adjust. Devise potential improvements.

The key tool of BP modeling is its graphical and symbolic interpretation called a flowchart or process map. These two terms can be used interchangeably but both stand for a type of diagram that represents a workflow of a process. MS Office suite also offers special shape objects to work with (Insert/Shapes/Flowchart) as shown in Figure 6. Each one of these standard sets of symbols has unambiguous meaning and is the subject of an ISO standard (ISO 5807:1985). [23]

### Flowchart



Figure 6: Flowchart symbols in MS Word

In the field of SW engineering, in the case of information modeling of organizations, we need to create a complex system of intertwined objects and relations. To assist with such complexity, methods, standards, and tools were developed to simplify and maintain order and prevent errors. Examples of these methodologies are described in the following chapters. SW tools supporting these standards and methods are called CASE (Computer Aided Systems Engineering) and various desktop or browser applications are broadly available. [8][24]

The following Chapters will introduce some of these BP modeling tools.

#### 1.3.1 ARIS

Architecture of Integrated Information Systems (ARIS) is a methodology for ensuring that Enterprise Information System (EIS) or any Information System

(IS) is properly designed to meet its requirements and will be working as intended. Compared to other frameworks (introduced in the following chapters), ARIS is rather a methodology, not a standard. This framework's strength is the ability to describe parts of the model through different optics or views. This five-view architecture is called the "ARIS house" and is depicted in Figure 7. This serves as a structure for a more detailed diagram illustrating ties throughout the enterprise (see Attachment 1): [16][18][25]

- **Organization view:** Describes the structure of a company, its employees and units, and the links and relations between them.
- **Data view:** Consists of events and states. Events define the changes in states of all company-relevant information objects.
- **Process view:** Is connecting all other views together. In this methodology, the processes work as a central integrating element.
- **Function view:** Contains functions, goals, sub-functions, and activities logically structured with hierarchical relationships.
- **Product/Service view:** Displays the company's portfolio of products or services. Sometimes might contain performance measures and metrics focused on continuous improvement.

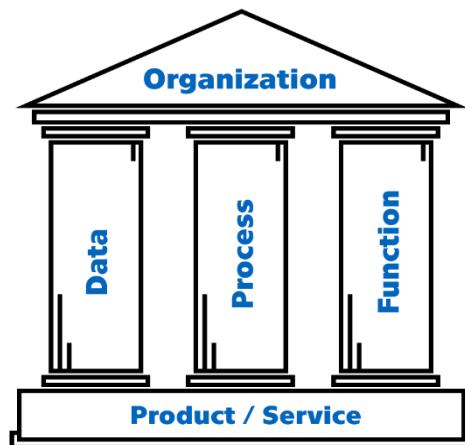


Figure 7: ARIS House. Based on [25]

This framework was formulated by Professor A. W. Scheer who also started SW company IDS Scheer, later acquired by Software AG, that developed a toolset called ARIS Express. This SW tool combines ARIS methodology with other frameworks and standards such as Event-driven process chain, UML or BPMN (see following chapters). The double-edged sword of this tool is its strong bond to the SAP ERP system since both originated from German-

speaking countries. It is well integrated and ensures good functionality within SAP, but also only with SAP and no other ERPs. [6][16][25]

### 1.3.2 UML

Unified Modeling Language (UML) is a general-purpose modeling language and standard using an object-oriented approach. It consists of an integrated set of diagrams of various agreed forms and shapes designed to help system developers to define, design, visualize and document SW system artifacts or business models.

UML was created in the 1990s by uniting multiple object-oriented methods and notations. The language was primarily intended for developing large SW systems, but thanks to using abstraction it is a suitable tool for modeling any system. In 1997, Object Management Group (OMG) adopted UML as a standard and in 2005 new upgraded version, UML 2.0 was published, following adoption by ISO (ISO/IEC 19505-2:2012). This standard is said to be methodology-independent which widespread its universality. [18][26][27]

For any system to be competitive, its modules must be structured in a way to allow scalability, security, and robustness. Architecture must be defined clearly enough for maintenance and effectively enough to enable execution under stressful conditions. UML standard offers a set of best engineering practices that are effective for modeling large and complex systems. To achieve effective system design, UML uses graphical notations or diagrams to express project or system design. UML 2.0 defines fourteen types of diagrams, divided into three groups: [8][16][27][28]

- 1) **Structure Diagrams:** describes static application structure of the objects, including their attributes, relationships, and operations:
  - a) Class Diagram
  - b) Object Diagram
  - c) Component Diagram
  - d) Composite Structure Diagram
  - e) Package Diagram
  - f) Profile Diagram
  - g) Deployment Diagram
- 2) **Behaviour Diagrams:** represent general types of behaviour and are the dynamic aspect of the system:
  - a) Use Case Diagram

b) **Activity Diagram**: is a graphical representation of the workflow, and thus is used to display the sequence of step-by-step activities from a start point to the finish point detailing various decision paths that occur in the progress. Its notation elements are shown in Figure 8.

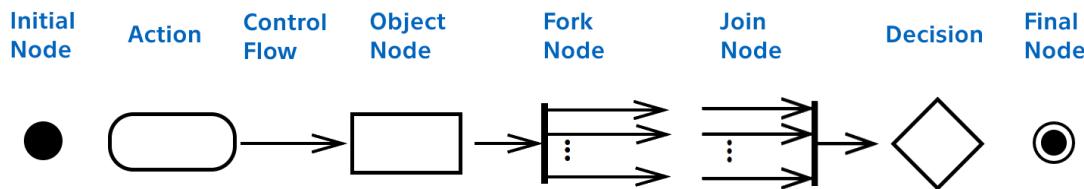


Figure 8: Notation for UML Activity diagram [28]

- c) State Machine Diagram
- 3) **Interaction Diagrams**: are a subset of behaviour diagrams and represent different aspects of interactions. They also describe the flow of data and control:
  - a) Sequence Diagram
  - b) Communication Diagram
  - c) Timing Diagram
  - d) Interaction Overview Diagram

Although UML and its diagrams are a powerful tool for SW development, in this thesis I am only briefly describing the Activity diagram as it is the most relevant UML diagram for process mapping and is commonly used for business modeling. Figure 8 is showing graphical notation symbols of the Activity diagram. Examples of UML diagrams are to be found in Attachments 2 (Activity diagram), 3 (Class diagram), and 4 (Sequence diagram).

### 1.3.3 BPMN

Business Process Model and Notation (BPMN) is a set of graphic objects and rules, which allow the modeling of processes. This graphical notation is presented and widely accepted as a standard BP modeling language.

The objective of BPMN is to support BPM, for both technical developers (responsible for implementing the technology that will support the performance of BPs) and business users who will manage and monitor these BPs), by providing a notation that is legible, intuitive, and unambiguous to all participants in the process life cycle, yet able to represent complex process

semantics. BPMN allows the creation of end-to-end BPs and is designed to cover many types of modeling tasks concerning business processes.

Behind BPMN creation stands BPMI (Business Process Management Initiative) and version 1.0 was introduced in 2004. Being based on the revision of other notations and methodologies and flowcharting techniques very similar to activity diagrams from UML. In 2005 BPMI was merged with the OMG (same group that stands behind UML) standard ratified, revised, and in 2011 the current version BPMN 2.0 was released. [8][12][30]

The main tool of BPMN is Business Process Diagram (BPD) or Collaboration Diagram. The structuring graphical elements of this notation will allow the viewer to differentiate between sections of a BPD. These elements are divided into 5 main groups with subdivisions: [30][31]

**1) Flow Objects:** these are the main graphical elements to define the behaviour of a BP and are related to the flow of information in the process. There are 3 core subclasses:

- a) **Events:** represent something, that happens during a process. These Events affect the flow of the model and usually have a cause (trigger) or an impact (result). Events are portrayed by a circle. Based on when an event affects the flow, we differentiate 3 types: Start, Intermediate, and End.



Figure 9: BPMN 2.0 Event types [31]

Within the circle, an icon denotes the type of event. Each such type can also be classified as Catching (that catch a trigger - all Start Events and some Intermediate Events) or Throwing (that throw a result - all End Events and some Intermediate Events). Throwing events have their icon filled with black. The whole table of all variations of event types is shown in Attachment 5.

- b) **Activities:** is work that is performed within a BP. An activity is represented with a rounded-corner rectangle. In BPMN 2.0 are 4 different types of activities as shown in Figure 10.



Figure 10: BPMN 2.0 Activities [31]

The first type - tasks, are the most common and can have multiple of its own types, that specify the nature of action that is performed. One of the icons shown in Figure 11 is then displayed in the top left corner of the task body rectangle.

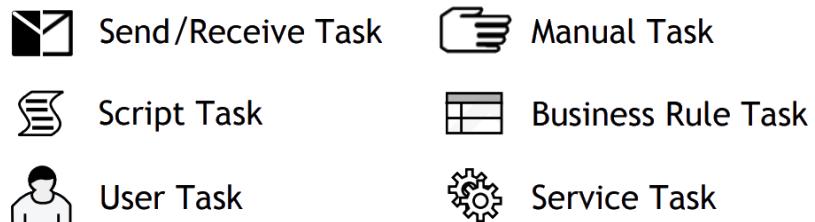


Figure 11: BPMN 2.0 Task types [31]

All activity types can also have markers, that are displayed in the bottom center. Not every activity can have all the markers. The most common is a sub-process that is used with tasks and call activities. Activity markers are shown in Figure 12.



Figure 12: BPMN 2.0 Activity markers [31]

- c) **Gateways:** are represented with a diamond shape and are used to control how sequence flows and interacts within a Process. A gateway is a decision point where, depending on the expressed conditions, is determined, or adjusted the path of a BP flow. Figure 13 shows the various types of gateways.



Figure 13: BPMN 2.0 Gateway types [30]

- 2) **Data:** In addition to capturing physical items, that are created, manipulated, and used during the execution of a process, we also need to capture the same for data. BPMN 2.0 uses graphical elements shown in Figure 14 for the representation of data (documents, communication, raw machine data) and its collection, reading & writing, inputs & outputs, and most importantly its flow:

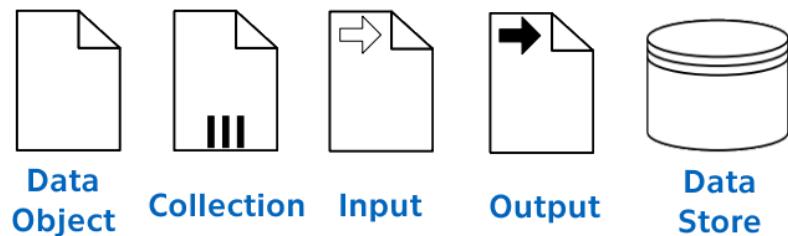


Figure 14: BPMN 2.0 Data elements [30]

- 3) **Connecting Objects:** unsurprisingly connect other objects together. It serves to show the succession and direction of the BP flow. BPMN differentiates the following types of connecting objects:

- a) **Sequence Flow:** is used to show the order in which activities will be performed in a process. The solid line can have a little diamond at its start, which indicates Conditional Flow (represents flow if a certain condition is met), or a diagonal slash, which indicates Default Flow as shown in Figure 15.

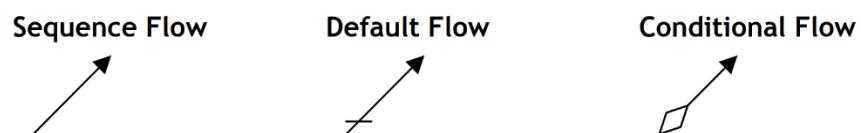


Figure 15: BPMN 2.0 Sequence Flow types [31]

- b) **Message Flow:** is used to show the flow of messages between two participants (2 separate pools) that are prepared to send

and receive them. Is represented by a dashed line with an open arrowhead at the end and an open circle at the start.

- c) **Association:** is used to link Artifacts and information, like text, with BPMN graphical elements and is represented by a dotted line. Sometimes can lack the arrowhead indicating direction.
- d) **Data Association:** is a special type of Association and uses the same notation. Is used to associate and connect data elements to Activities, Processes, and global Tasks. Figure 16 shows the Message Flow and both Associations.



Figure 16: BPMN 2.0 Connecting Objects [30]

- 4) **Pools & Swimlanes:** A Pool is the graphical representation of a participant in a BPD. A Swimlane is a sub-partition within a pool used to categorize graphical elements according to their function or role. An example of a pool with two lanes is shown in Figure 17.

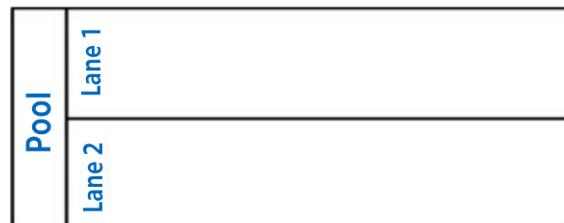


Figure 17: BPMN 2.0 Pool with 2 Swimlanes [12]

Sharp and McDermott are stressing the importance of neat organization of elements within the pools and swimlanes: "Sequence and dependency (time) flow from left to right" because the purpose of a workflow model is to show the real flow of work: [5]

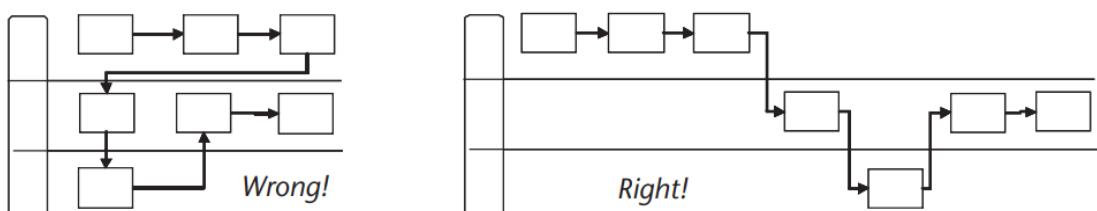


Figure 18: Correct way to capture workflow within the pool [5]

5) **Artifacts:** allow to bring additional information about the process into BPD. Artifacts do not affect the flow of the process. Modelers or modeling tools are free to add as many Artifacts as necessary but only two are standardized Artifacts shown in Figure 19.

- a) **Group:** is a visual grouping of elements that are within the same logical category. Such objects are encapsulated with a dot-dashed line and labeled with the name of the Group.
- b) **Text Annotation:** object can be connected to a specific object on the Diagram with an Association. Text Annotation is placed within the bounds of the open rectangle.

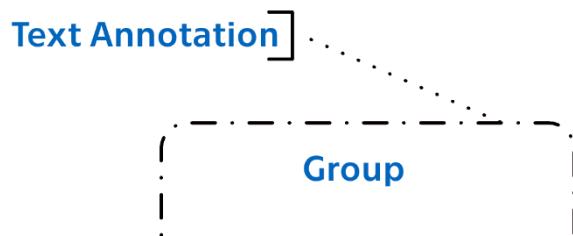


Figure 19: BPMN 2.0 Artifacts [31]

Outside BPD or Collaboration Diagram, BPMN 2.0 also has Choreography and Conversation diagrams. Those are enhancing the functionality of this notation in specific fields, that for complex examples can be difficult to depict in BPD and thus preventing errors and ensuring clarity. [30][31]

The process in BPMN 2.0 notation designed and modeled by me is presented in Chapter 8.2 as a solution to a particular problem.

When used properly, BPMN 2.0 is a very powerful tool. However, due to its complexity, it is prone to human error. This notation also has its shortcomings facing criticism. Examples of some missing features, that would be useful if BPMN reflected them, might be: [8][12]

- Resource requirements
- Performance metrics
- Process status
- Objectives & goals
- Locations

## 2 Barcodes

To increase the process effectivity barcodes can be utilized. In this Chapter, I will introduce what these barcodes are, how were developed and evolved, how they work and where they can be used.

Barcodes are defined as an optical machine-readable representation of data, that is formed by combinations of low and high reflectance regions of the surface of an object, which represents zeroes and ones in binary code. Barcodes usually carry the data and information related to the object to which it is attached. The most common is to use them for Unique Identification (UID) of any goods or products. Barcodes belong to a group of Automatic Identification and Data Capture (AIDC) technologies or methods, alongside RFID, Biometrics, Optical Character Recognition (OCR), or Magnetic stripes (see Chapter 4). With technological progress barcodes evolved into all 3 dimensions. For all such representatives, literature uses the word "Barcode", even though barcode is only one type (though the most common) of one-dimensional. In this thesis, I am using both, word code and barcode in the same meaning, even though all barcodes are rather a symbol than code by its literal definition. [32][33]



Figure 20: 2D vs 1D code [34]

- **One-dimensional (1D) codes:** also called linear codes encode the data in one dimension. The vertical dimension serves to ease reading. Such codes contain only a limited amount of data and thus work as an index to a backend database, that contains data about products. [33]
- **Two-dimensional (2D) codes:** called matrix codes contain data in both, horizontal and vertical dimensions as depicted in Figure 20. Since 2D codes can encode far more data, they can bear all required data and thus serve as portable databases. The increased capacity and density of information allow a much more versatile spectrum of applications compared to conventional 1D codes. [33][34]

- **Three-dimensional (3D) codes:** also called "bumpy barcodes", are 1D or 2D codes directly embossed on the surface of an object. Such codes use spatial elements to reflect high and low signals, rather than visual contrast of different colours on plane surfaces. Bumpy codes have the advantage of being coated with colour and still being readable. They are usually chemically etched, dot peened marked or laser engraved on the surface of a product. Despite literature calling bumpy codes 3D codes, I don't think it's a correct designation, since such codes do not contain data in the third dimension. [33][38]

## 2.1 Barcode Evolution

Though the attempts to create the AIDC system were undertaken already in the 19<sup>th</sup> century, it was in the 1950s when Woodland and Silver patented their "Bull's eye" barcode. Authors were incentivized by the Philadelphia food chain and their demand for automatic product information read during checkout. Woodland and Silver were said to be inspired by the Morse code and the round form of the barcode was redesigned later, only to make reading easier. [32][33]

In the 1960s system called KarTrak was adopted for US railroads. Using red, white, and blue horizontal bar labels on the sides of railroad cars and static optical scanners, served to read serial number information from moving railroad cars. This application entitles the two firsts: industrial barcode application and the use of different colours in a barcode. [33]

In 1968 David Collins, who worked on KarTrak, co-founded the first company to design and manufacture equipment for scanning using innovative helium-neon laser technology. The company invented Code 2 of 5 and then Interleaved 2 of 5 (ITF) encoding. System was used in 1972 for identifying engines and axles at General Motors factories. [33]

The birth of the modern 1D code is considered the year 1973 when the US grocery industry formally established the Universal Product Code (UPC) as the standard, based on IBM linear barcode developed by George Laurer. UPC barcodes are still widely used for Point of Sale (POS) systems. [33][35]

Inspired by UPC, which was designed to be used only within the US, European Article Numbering Association adopted a code format called European Article Number (EAN) in 1977. EAN took into consideration global usage and

compatibility with the UPC and thus is referred to as International Article Number (IAN) and is globally accepted (see Chapter 2.4.3). [33][36]

The United States Department of Defense adopted the Code 39 standard for automated marking and reading in logistics in 1981 for its suppliers and contractors. Code 39 can encode alphanumeric characters (collection of Latin letters and Arabic digits) and is still widely used in industry. The same year the denser variant was invented to be able to include all 128 characters of ASCII (American Standard Code for Information Interchange) and therefore called Code 128 (see Chapter 2.4.3). ASCII is an (originally 7-bit) character encoding standard for electronic binary communication. [33][37]

As the amount of data captured in one dimension is limited, efforts to enhance the capacity of barcodes emerged. The logical solution was to stack multiple 1D linear codes on top of each other. Stacked codes are an older type of 2D codes, and they make use of vertical dimensions to store additional data. Its advantage is its simplicity and sometimes the possibility to use the same reading technologies as for conventional 1D codes.

The second and younger type of 2D codes are matrix codes. Compared to stacked types, matrix 2D codes can encode far more data per the same area. Figure 21 depicts the Barcode evolution in relation to symbol capacity. Modern codes using multiple colours are developed to enhance the capacity even more. At the same time, reading technology must be more advanced to make effective use of such codes in similar reading times. Examples of both types of 2D codes are described in more detail in Chapter 3. [33]

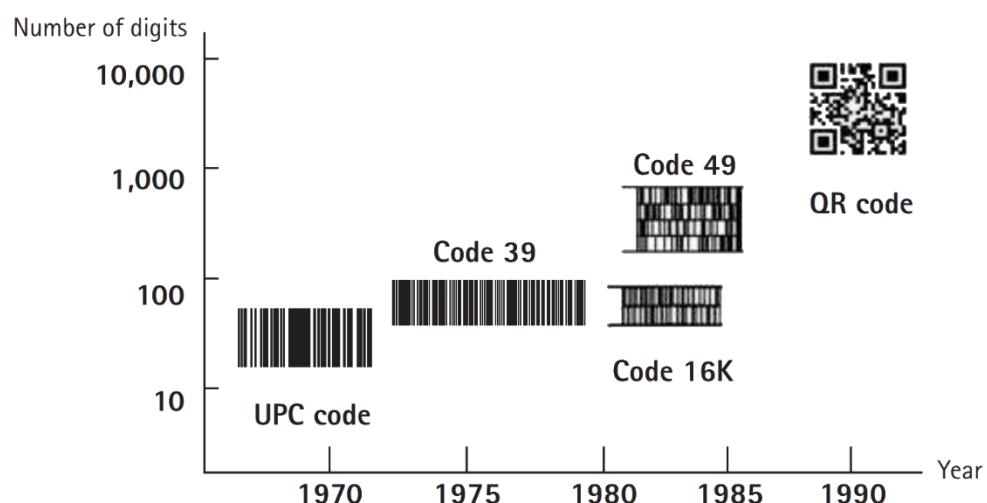


Figure 21: Barcode evolution [35]

## 2.2 Barcode Anatomy

Since the whole barcode terminology might be confusing, it is very important to differentiate between basic terms. For barcodes applies: [32][33]

- **Symbology:** is a specified set of rules for encoding data. into a 1D or 2D code format. Using the same rules in reverse we can read the barcode as decoded information.
- **Symbol:** is a graphical mark or sign, that is created by applying the symbology rules and is composed of contrasting symbol elements.

Both 2D and 1D barcodes have similar elements, though the two-dimensional ones tend to be more advanced. Such elements are: [33][35][36]

- 1) **Quiet zone:** is the area of high reflectance and helps detection of the code symbol, clearly separating the symbol from its surrounding.
- 2) **Finder elements:** are unique characters. They instruct the scanner when to start reading, in what direction to read, and when or where to stop. Start and stop characters, for 1D codes. For 2D codes following:
  - a) **Finder patterns:** (Cornerstones, Borders, Guide bars, etc...) Thanks to these elements position, size, and angle of the symbol can be determined. This enables 360° reading.
  - b) **Alignment Pattern:** bigger 2D codes use these marks for corrections in case of bulges and other distortions.
  - c) **Timing Pattern:** is arranged in both vertical and horizontal directions and is correcting the central coordinate of the data cell in case of error or distortion.
- 3) **Data elements:** is either an internal 2D area or a 1D axis filled with characters representing data, containing the information for users.
- 4) **Special elements:** these ensure proper recognizability of code
  - a) **Checksum:** 1D codes may perform a mathematical operation to ensure accurate decoding of the barcode.
  - b) **Masks:** some 2D codes have special patterns to have B&W cells well arranged in a balanced order for easier reading.
- 5) **Axis dimension:** is the width (width and height for 2D codes) of the smallest element (module) of a barcode. The smaller this element can be, the more data we can store. Naturally, there are limitations of contemporary reading technologies for ensuring clear recognition.

## 2.3 Barcode Readers

Barcode Reader (BR) is a device that makes use of the reflexivity of different colours to read and decode contained data. White colour reflects light the best, the black colour absorbs it the best. BR devices can also be called optical scanners. It usually has a light sensor for translating optical impulses into electrical signals. A decoder circuitry then analyses and sends the data through a port to a connected computing device. [32][36]

There are many forms and sizes of BR. Handheld scanners, Pen-type readers, static industrial camera gates, or even cameras in our smartphones. I will introduce two main technologies behind the functionality of BRs. [33][38]

### 2.3.1 Laser Scanners

The laser beam is used as a light source and the ray goes through oscillating mirrors or rotating prisms to scanned objects. A photodiode sensor then measures the reflected rays from the object and inducts an analog signal, that is further converted into a digital signal. Laser scanners are the older technology of the two, designed for reading 1D codes. But it still has its benefits. They are fast, do not require an image processor, and with the use of special optics they can read from relatively long distances. Disadvantages are troublesome reading poorly printed, low contrast, distorted, or damaged 1D codes. Laser scanners also have moving parts, which can break, resulting in additional costs. Finally, they also need to be shielded due to eye safety concerns. [32][38]

### 2.3.2 Image-Based Readers

Such readers employ one of two similar technologies. The first is called Charge-Coupled Device (CCD), and it is an image sensor. This electrical integrated circuit is an array of capacitors that is capable of converting light input into an electronic signal. This technology is widely used in CCD scanners, that contain Light-Emitting Diode (LED) that emits light to a symbol, mirror, and lens to guide reflected light to the CCD sensor, and finally processing circuitry. CCD sensor can be either linear for reading 1D codes, or an area sensor for reading 2D codes. Such scanners are cheaper and more durable than their laser counterparts. Older digital or phone cameras also use CCD sensors.

Complementary Metal-Oxide Semiconductor (CMOS) sensors took place of its predecessor CCD in most commercial image-based technologies. CMOS has multiple advantages. It has a faster data rate because the pixel signals can be accessed directly and sequentially and use less power. Both types of BR are shown in Figure 22. [33][38][39]

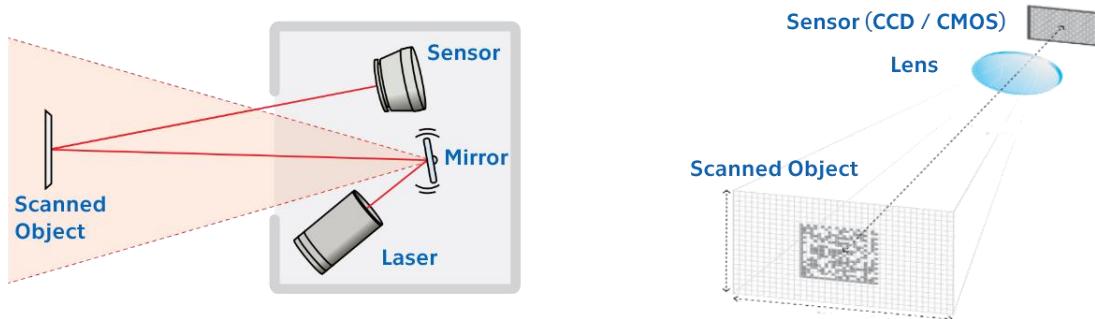


Figure 22: Laser & Image-based readers [38]

With technological progress and its achievements, most of us are daily using devices (smartphones, laptops, tablets) that are capable of reading barcodes if suitable SW is installed. Following [QR2](#) is linked to an android app called Scandit Barcode Scanner Demo available on Google Play for free. I recommend installing this application since it allows you to scan most of the barcodes there are. The application is also applicable and useful for reading this thesis and its examples.



QR 2: Scandit Demo – BR Mobile Application [40]

## 2.4 Symbology of 1D Codes

Five various types of 1D codes and their symbology are standardized by ISO: Code 39, ITF, EAN/UPC, Code 128, and the GS1 DataBar family. GS1 is an international organization that develops global standards for business and supply chain communications (including barcodes). GS1 acts as a global collaboration platform that brings various stakeholders together to develop solutions for challenges of data exchange. Using GS1 standards, 6 billion transactions are carried out every day. [33][42]

This Chapter introduces the classification of these widely used 1D symbologies by 4 pairs of factors described below and shown in the following Figure 23. And following Chapters then introduce 3 representatives of these 1D barcode symbologies in deeper detail: [33][41]

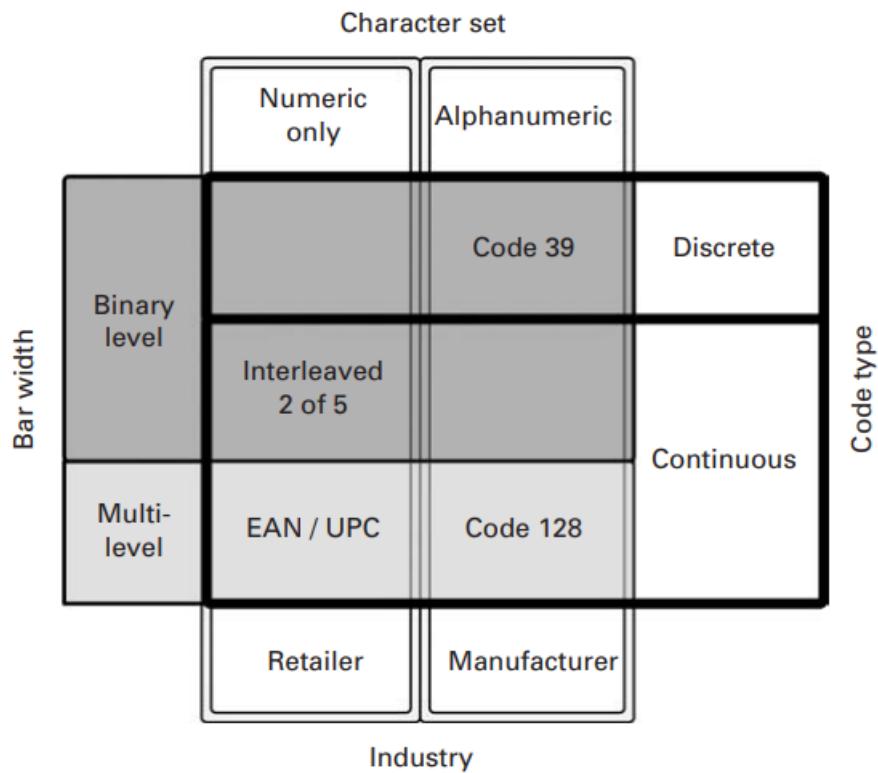


Figure 23: Classification of 1D code symbologies [33]

- A character set is either **Numeric** (consisting of Arabic digits from 0 to 9) or **Alphanumeric**, which includes Arabic digits and Latin letters.
- **Discrete** code type includes additional space between characters, that does not convey any information. This makes each code character distinctively identified but makes code more spacious compared to

the **Continuous** code type, where each character starts with a bar and ends with a gap or vice versa achieving a higher data density.

- **Binary** bar width symbology employs only “wide” and “narrow” bars and gaps, contrary to **Multi-level** bar width symbology, which can have more than 2 widths of bars and gaps. They are referred to as  $(n, k)$ , where  $n$  is the number of modules for each character and  $k$  indicates the number of basic width units.
- Industry states the most common field of usage of specific symbology. Retail usually uses just numeric symbologies, thanks to employing index - database - POS systems.

#### 2.4.1 Code 39

Developed in 1974 by Intermec, Code 39’s name comes from the fact that it originally could encode 39 characters. It is a variable-length, binary-level (only narrow and wide bars and gaps) discrete (Data Characters (DC) are separated by N gap) barcode symbology. Each DC consists of various combinations of 5 bars and 4 gaps. Character set includes A-Z Latin uppercase letters, 0-9 Arabic digits, 7 special characters (-, ., \$, /, +, %, and space). An additional character (asterisk = \*) is used for both the start and stop character. Figure 24 shows the Code 39 symbol and its features:

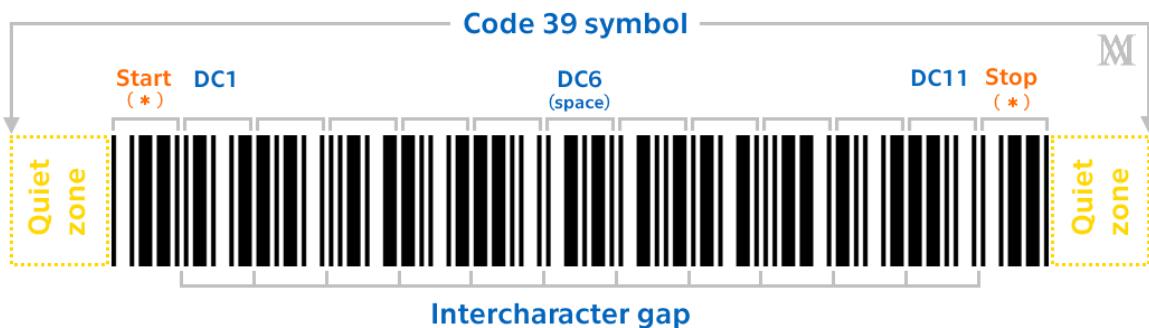


Figure 24: Code 39 symbol. Based on [33]

Code 39 does not include a Checksum, but it is considered self-checking. The biggest drawback is low data density. This barcode is commonly used in sectors such as government/defense, manufacturing, logistics, or medical. To generate barcode examples, I used online browser generators, but MS Excel can operate with special fonts creating barcodes as well. [33][41][43]

## 2.4.2 UPC / EAN / IAN

As already mentioned in Chapter 2.1, EAN (sometimes called IAN) code format is based on UPC symbology and is its superset, therefore any SW or HW capable of reading EAN symbols should automatically be able to read UPC symbols as well. There are many variations of both, EAN and UPC codes. Most commonly used are UPC-A and EAN-13. Smaller versions of both (containing fewer DCs) are labeled as UPC-E and EAN-8 respectively and are used on smaller products or packages, where their bigger counterparts would not fit. Since UPC codes are used in the USA and Canada and EAN codes globally, the latter is introduced in more detail. [33][41]

### EAN-13

Created as a superstructure of UPC-A by adding one more digit, EAN-13 is a fixed-length 13-digit (12 data and 1 check) barcoding standard. It can encode only numeric characters and belongs to multi-level, continuous barcodes. Each DC is consisting of two bars and two spaces a total of seven modules. The widths of the bars and gaps vary between one and four modules. So, in the  $(n, k)$  notation mentioned earlier, EAN-13 would be  $(7, 4)$ .

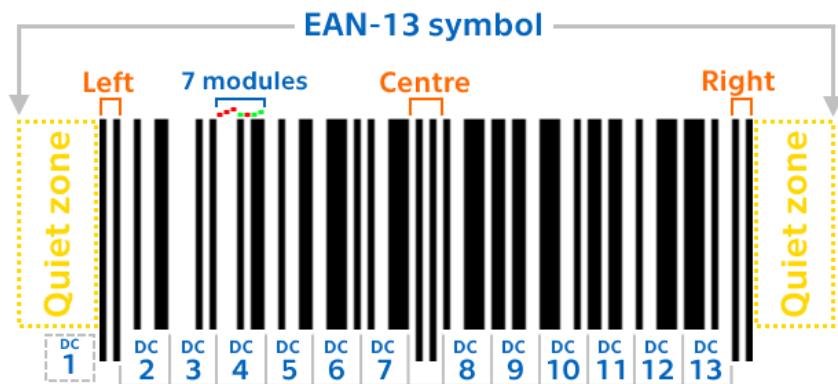


Figure 25: EAN-13 symbol. Based on [33]

The EAN-13 symbol shown in Figure 25 starts and ends with guards consisting of bar-gap-bar (1-0-1) of a total module length of 3. In the center, there is a similar guard (0-1-0-1-0) that separates the left and right halves with DCs. The right half DCs of the symbol is encoded using an even-parity encoding pattern. The left half DCs can be encoded using two different (odd and even, 3 of each) left-hand encoding patterns. Based on the combination of these parity patterns, added the first digit is calculated (it is not represented directly by a physical pattern but is encoded indirectly). If the first digit is 0, then the rest of the EAN-13 code is identical to the UPC-A code.

Different encoding patterns for left and right sides enable reading codes even when scanned upside down. The first three digits (DC1-DC3) represent the country code or so-called GS1 prefix. DC4-DC7 represents the registered manufacturer. DC8-DC12 represents the product code and last, DC13 is the Checksum digit. [33][36][41][43]

EAN-13 is the most used barcode in retail as a Global Trade Item Number (GTIN), which also incorporates other standards such as UPC or International Standard Book Number (ISBN) and many others. [44]

### 2.4.3 Code 128

Named after the number of characters of ASCII, that can encode, Code 128 is a variable-length multi-level, continuous symbology. It requires the smallest area to encode data of 6 DCs or more of all the 1D symbologies due to the use of a unique compression technique. ASCII characters are separated into 3 subsets A, B, and C and each has a special Start character to indicate which subset will be used. These can change within the progression of code. Each DC composes of 3 bars and 3 gaps, the width of which varies from one to four modules. The sum of the widths of the gaps must be odd and the sum of the widths of bars must be even, totaling 11 modules for each DC (apart Stop pattern character, which has 13 modules). Therefore the  $(n, k)$  notation for Code 128 is  $(11, 4)$ . The last DC is a Checksum as shown in Figure 26. [33][41][43]

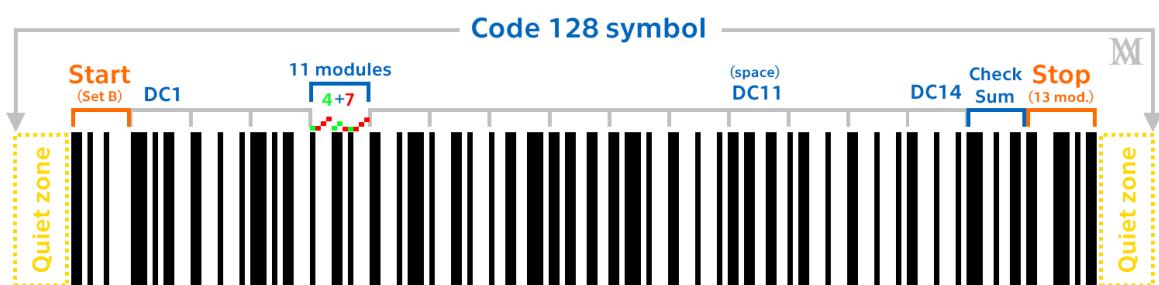


Figure 26: Code 128 symbol. Based on [33]

This barcode is widely used in logistics (transport, shipping, tracking), usually labeling boxes and packages with products (which themselves have EAN codes for retail POS identification) inside. Its advantages are the ability to encode almost any character and high data density. [33][41]



### Iceland Example 1: PCR testing

The people of Iceland and their government managed the COVID-19 pandemic with reason and science. Emphasis on personal responsibility without unnecessary restrictions. Sick pay, quarantine hotels, voluntary vaccinations, and free testing for citizens and even foreigners. All of this prevented any massive outbreak of the disease in this country. [45]

I was tested several times and the process flowed as follows:

After filling out a form on the governmental website covid.is the case was created in their database and a code was generated and assigned to it. Barcode (Code 128 in Figure 27) representing this case was sent to my email with the invitation stating the location, date, and time of the testing. At the health facility, I showed the barcode on my phone screen, and it was scanned, printed, and labeled on the test swabbing vial. I continued to the sampling area and gave the vial to the testing personnel. After testing, they sent the sample in the vial to a laboratory, where it was scanned again, and the sample was automatically assigned to my case in the database. Within 24 hours I received an automatic email with the result and further instructions and recommendations if positive.

PCR TEST



Figure 27: Barcode I received for PCR testing



## 3 Two-Dimensional Codes

How and from what 2D codes evolved was introduced in the previous chapter. In this chapter, I will introduce a mathematical-based element, that enabled 2D codes to withstand a certain degree of damage - Error Correction (EC). Then I will introduce the most common 2D code types and examples and describe in what fields and how they are and can be utilized.

### 3.1 Error Correction

For 1D codes, which have lower capacities, the Checksum element is sufficient. Some 1D codes don't even have Checksum and if the barcode symbol is damaged, we can still obtain the data by transcribing them from below the symbol, where usually the human-readable data are displayed.

Low capacity 2D codes are limited by the choices of possible EC codes. Therefore, codes that require less capacity, such as parity-check or Hamming codes (linear EC code, invented by R. Hamming in the 1950s) are being used.

Contrary to 1D codes, 2D codes contain vast amounts of data storage, and hence can't be displayed in the human-readable form below the symbol. Because of this vastness, one mistake in the data area could inflict the unreadability of the symbol. Luckily, 2D codes are redundant in capacity and can store so-called EC Codewords, which help to detect errors and restore some amount of data if damaged. [33][47]

#### Reed-Solomon codes

Introduced in 1960 by I. Reed and G. Solomon, these codes can detect and correct errors in data. Reed-Solomon (RS) codes are based on and are a subset of Bose-Chaudhuri-Hocquenghem cyclic EC codes (BCH codes). RS uses the "magic of math" including Finite (Galois) Fields, Modular Arithmetic, Lagrange Interpolation, and all kinds of matrices – mathematics too complicated for common mortal. Due to this complexity RS EC codes are computationally intensive. Despite that RS EC is used by almost every 2D code symbology. [33][47]

## 3.2 Stacked 2D codes

### 3.2.1 PDF 417

Is the most commonly used stacked linear 2D code. It was invented by Ynjiun P. Wang in 1991 and is now standardized by ISO. PDF 417 uses multi-row, continuous, variable-size symbology. PDF stands for Portable Data File and number 417 represents 17 modules for each Codeword (CW), consisting of 4 bars and 4 gaps of variable widths. These CWs represent a number from 0-928 (929 encoding), 900 of which can be used for data and the rest are for special functions. Each CW can be symbolized by 3 different patterns or clusters. CWs in the same row are using the same cluster encoding. These three clusters are called Cluster 0, Cluster 3, and Cluster 6 and are repeatedly used in sequence for rows. PDF 417 symbol consists of Start and Stop pattern (uniquely has 5 bars), Left and Right row indicator. Data columns and rows with CWs. Symbology is capable to encode in 3 predefined modes: byte, text, and numeric. This allows encoding more characters into one CW. In byte mode it is 1,2 bytes per CW, in text mode, it is 2 characters per CW, and in the numeric mode, it can represent 2,9 digits per CW. This is exactly the reason why the term CW should not be interchanged with DC. [33][36][41][43]

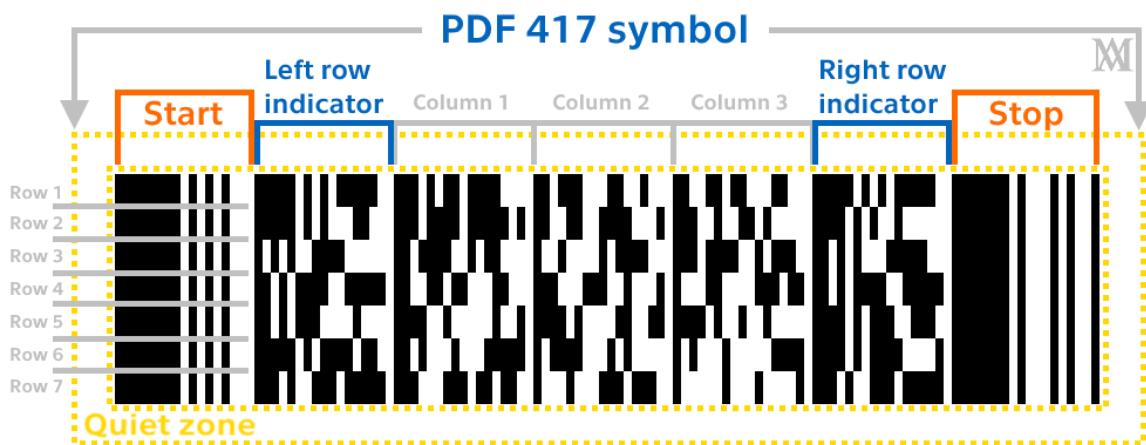


Figure 28: PDF 417 symbol. Based on [33][36]

The symbol in Figure 28 contains 23 letters but consists of only 21 CWs (3 columns and 7 rows). Most of those are for carrying data but at least two of those are for Error Correction (EC). PDF 417 uses the RS EC technique, which allows correct data reading despite minor damage to the symbol. The more EC CWs symbol contains, the more robust and resilient to damage it becomes but contrary there will be less space for data. In Attachment 6 the

detail of the first CW (column 1, row 1) and its 8-digit bar-gap sequence number (31112144) and calculation check for Cluster number can be found. PDF 417 is a public domain, free to use for everyone. [33][41][46]

If you ever used an airplane, you could find PDF 417 on your boarding pass. Next to transportation and logistics, this 2D code is also used in the government sector for identification. In many states of the USA serves to encode drivers' key information on the backside of driver's licenses. [33][41]

### 3.3 Matrix 2D Codes

#### 3.3.1 MaxiCode

Developed for the purposes of United Parcel Service (UPS) in 1992. MaxiCode is fixed size matrix 2D code. Inch by inch in area and has the possibility to chain data with other 8 MaxiCodes to convey more data. Being specially developed for parcel logistics, it utilizes Structured Carrier Message (SCM) data format, regarding standardized key parcel attributes. It is unique by using hexagonal modules that create an array of 886 modules in 33 rows containing 30 and 29 modules alternately, arranged around a bulls-eye finder pattern. This innovative element enabled the high-speed reading of labels on packages moving on conveyor belts. Orientation patterns take 6\*3 modules, and 4 modules are reserved for mode selection as depicted in Figure 29. The remaining 864 data modules create 144 6-bit CWs. [33][43][48]

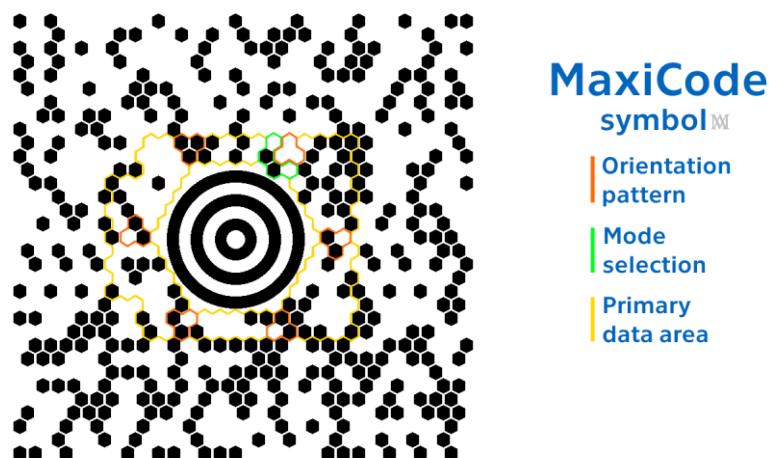


Figure 29: MaxiCode symbol. Based on [33][48]

Having 7 different modes, we can set up whether SCM format or general data will be encoded, whether alphanumeric or just numeric are stored, and what type of RS EC will be applied. There is Standard Error Correction (SEC) or

Enhanced Error Correction (EEC). The former is able to restore all data even when 25% of all modules are destroyed. Data in SCM format are stored in either primary or secondary message components. Former contains the selected mode, postal code, ISO country code, and class of service. The latter contains the tracking number, package weight, shipping address, and other, supplemental information. Despite its specialization in tracking and managing shipments of packages and its lower capacity compared to other matrix 2Dcodes, MaxiCode can be used for anything, where fast scanning is required and where a risk of damage to the label exists. [33][41][48]

### 3.3.2 Aztec Code

In 1995 Andrew Longacre and Robert Hussey invented "Two-dimensional data encoding structure and symbology for use with optical readers". From an orthographic top view, the square bull's eye finder pattern in the center resembles Aztec temples, after which the symbol is named. This "pyramid" consists of middle black dot and 3 black square rings interspaced with 3 white ones. In each corner of the finder pattern are 3 modules (or bits) serving for detecting the orientation of the symbol. From the middle of each side of "the pyramid" the Reference grid is spreading out. This grid is always alternating white and black bits and horizontal and vertical components are added at every 16<sup>th</sup> bit towards the edges of the symbol. This grid structure enables correct distortion in larger symbols. Finally, the Mode message takes 40 bits of data alongside the finder pattern fitted in between the orientation pattern. These bits store information about the number of layers (and thus about the size of the symbol) and the length of the message. Figure 30 depicts the Aztec Code symbol and its features: [43][49][50]

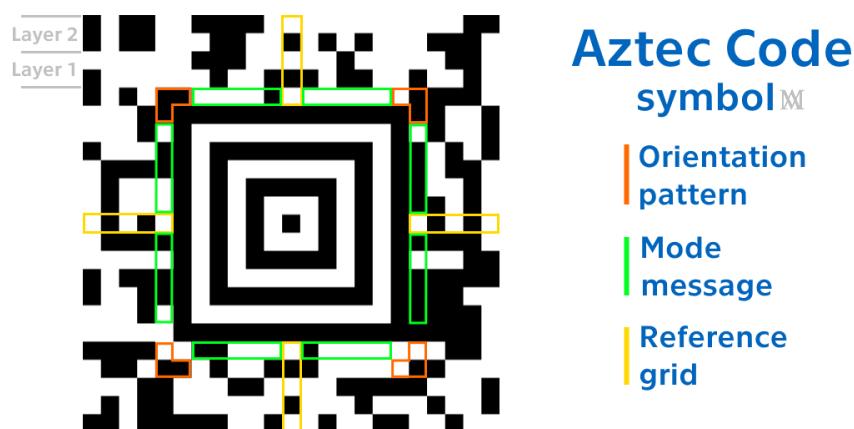


Figure 30: Aztec Code symbol. Based on [43][50]

Data are stored in layers, that are always 2 bits thick and the CWs length depends on how many layers of data there are (detail of CWs scaling based on a number of layers, their orientation, and Reference grid are displayed in Attachment 7). Since Aztec Code is a variable-size symbology, the symbol itself can vary from 15x15 to 151x151 modules in size. [43][49][50]

Aztec symbology supports all 255 characters and uses RS EC. It is also standardized by ISO and the public domain, despite being patented. It is said to be advantageous for space-saving, since it does not need a quiet zone, but current reading devices seem not to be bothered by disruptions in quiet zones. Aztec Codes are vastly used in transport ticketing. [36][41][50]



### Iceland Example 2: Bus Ticketing

An unnamed transport company operating buses in Reykjavik was using its own developed mobile app for single fare tickets, with the timer countdown displaying the validity of the ticket. Since buses in Iceland are relatively expensive, many people tried to cheat on this system. Some would use a screen-captured video of the countdown to imitate the valid ticket in the company's app. This resulted in a need for better verification, thus enhancing the apps function to display video camera capture in the background of the countdown timer when showing to the bus driver as shown in Figure 31:

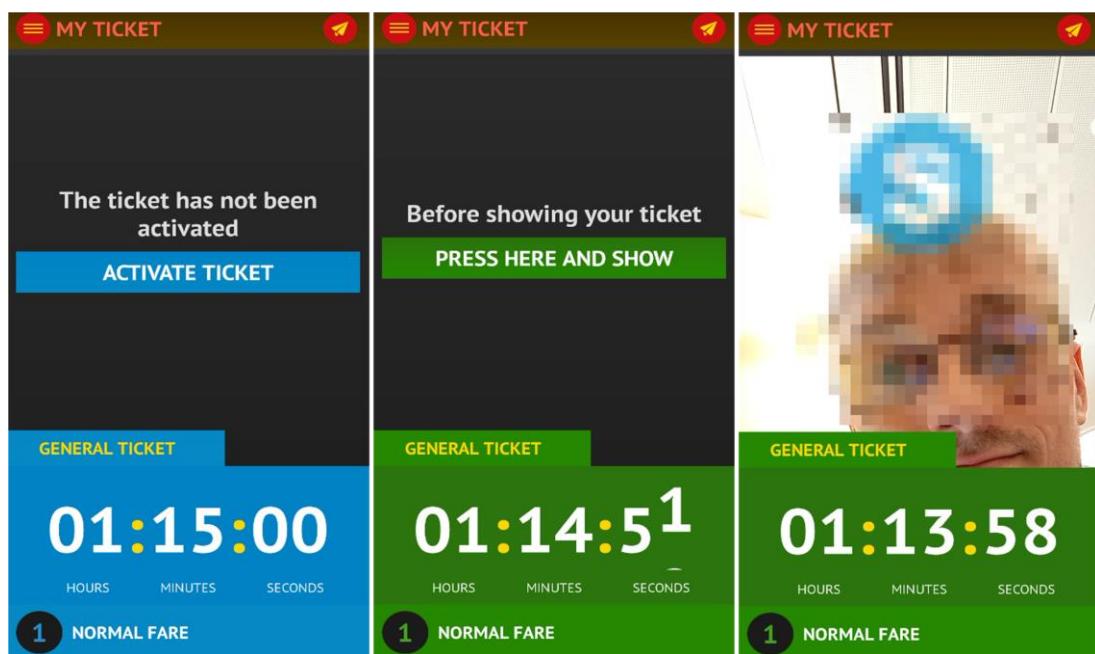


Figure 31: Old bus ticket verification

Despite this solution revealing fraudsters, it was still an ineffective and obsolete ticketing system. That is why the company joined forces with another developer that would implement Aztec Code ticket validation and create a more reliable application. This step required to install new optic readers on every bus. The app enables smoother purchasing of various tickets and storing them in your mobile app. If you have a valid ticket, the app will generate an Aztec Code, that is to be read when entering the bus. This symbol is regenerated every 11 seconds and contains a human unreadable code of 519 characters. The validating system requires an internet connection on both sides and utilizes asymmetric cryptography when communicating with the ticket validation database on the operator's side. An enhanced example of this new encrypted ticket and its decoded "text" is depicted in Figure 32. [51]



Figure 32: New bus ticket verification.

Although this solution is very safe and robust against any fraud, the scanning is done when entering the bus by the front door, so the inflow speed of new passengers remains the same as with the old system.



### 3.3.3 HCCB

High Capacity Color Barcode (HCCB) was developed in Microsoft by Gavin Jancke in 2007. It utilizes a color palette to enhance the barcode capacity. So do other barcodes such as Korean ColorCode or JAB Code. But the HCCB also uses a unique triangular module design to further increase capacity per area. Thanks to these features, HCCB can achieve three times the density of industrial-standard 2D codes in the same given space. The structure of the symbol consists of triangular cells put in rows, with white separators between adjacent modules to enhance the accuracy detection. Different variants are using 8, 4, or just B&W colors as shown in Figure 33. The symbol can be printed using a regular ink or laser printer either in greyscale or color tones. Incorporation of a reference palette within the symbol plus robust RS EC allows performing correct reading even on symbols whose colors faded due to aging, smeared, blurred, or otherwise damaged. [33][52][53]



Figure 33: HCCB types and comparison [52]

Code was originally intended for identifying commercial audio-visual works within Microsoft Store such as motion pictures and video games. Due to high data density, HCCB symbols can be used for identification or transaction applications, where great overall information content is demanded. Because of its uncommonality, not a lot of scanners can read this code. The same applies to the mobile app Scandit, which I recommended earlier. [33][52]

### 3.3.4 Data Matrix

Data Matrix (DM) is one of the most widespread 2D codes. The current upgraded version Data Matrix ECC 200 replaced its older versions ECC 000-140 in 1995. In 2000, DM was accepted as an ISO standard and is also approved as an industrial standard by GS1 and other organizations. [33][54]

DM symbol always has an even number of modules on each side. Size can vary from 10x10 up to 144x144. Though the maximum size of one block is 24x24 modules and bigger symbols are just multiple blocks put together. DM ECC 200 uses robust RS EC and can sustain a high amount of pixel damage. The symbol itself has 2 solid border sides serving as finder and orientation pattern known as the "L pattern". The other side with a broken pattern is called "Clock Track" and determines the size and distortion of the symbol. The inside area of the matrix consists of pixels clustered in 8-bit Codewords. The pattern of CWs starts at the top left corner and by the sides, some CWs are divided into 2 parts. Also, there are 4 cases of arrangement of pixels in the remaining 3 corners depending on the size of the symbol. Sequences of 8-bit CWs (MSB/LSB stands for the Most/Least Significant Bit) and other important features are depicted in the following Figure 34. [33][43][54]

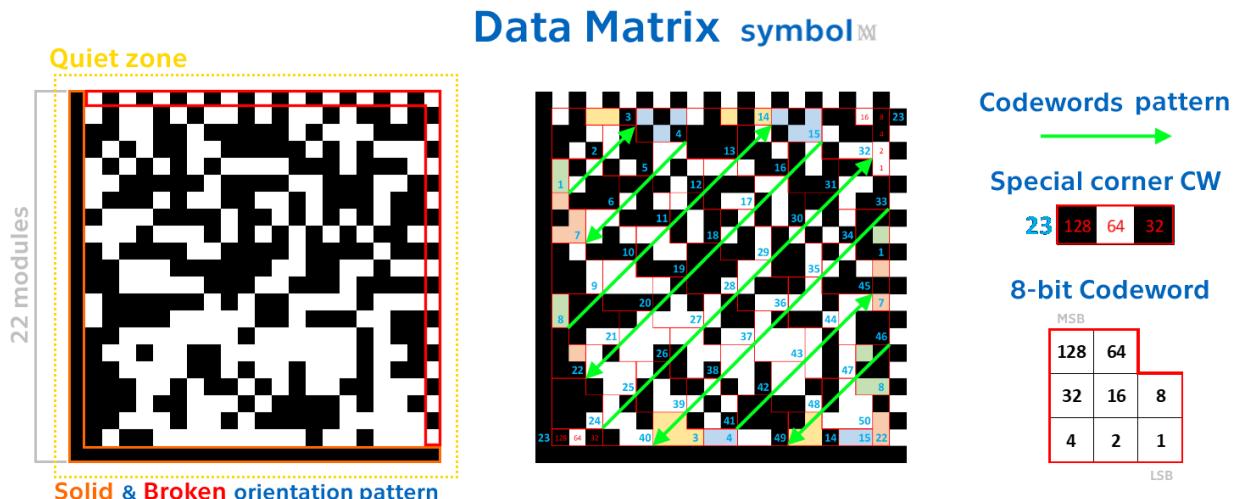


Figure 34: Data Matrix symbol. Based on [33][54]

DM in Figure 34 is 22x22 modules in size and thus contains 30 CW of data and 20 CW for EC. The first CW has a value of 230 which sets encoding for the C40 mode, suitable for upper case text. The maximum size variant can contain up to 2335 alphanumeric characters or 3116 digits. Other advantageous features are the ability to invert colors for printing on dark surfaces (Scandit app cannot read inverted symbols), or structured append, which allows to fill up to 16 DM symbols with continuous message readable regardless of the scanning order. DM can generate very small symbols which make it perfect for use in marking little parts for example in manufacturing, medical, aviation, or electronics industries. The symbols can be as small as 2.5x2.5 mm. An example of such little code engraved on pharmaceutical vials can be found in Attachment 8. [33][36][41][55]

### 3.3.5 QR Code

In 1994 Denso Wave, a subsidiary of automotive Toyota's supplier developed a 2D barcode focused on high-speed omnidirectional component reading. Named after this ability, Quick Response (QR) Code quickly became a favorite throughout many industry sectors for its great data storage capacity. Standardized by ISO and being a public domain, this symbol is ubiquitous.

Model 2 of QR Code, formulated in 2006 is commonly used today. Its predecessors lacked an Alignment pattern. There is also a Micro QR variant, designed for smaller amounts of data on limited areas. The QR symbol can be generated in multiple versions, depending on the amount of data to be encoded. The smaller Version 1 consists of 21x21 modules and can contain up to 41 numeric or 25 Alphanumeric characters. The biggest symbol – Version 40 consists of 177x177 modules, and the maximum capacity is 7089 numeric or 4296 Alphanumeric characters. [33][41][56]

Figure 35 shows Version 4 (33x33) symbol, and its main features. [33][43][56]

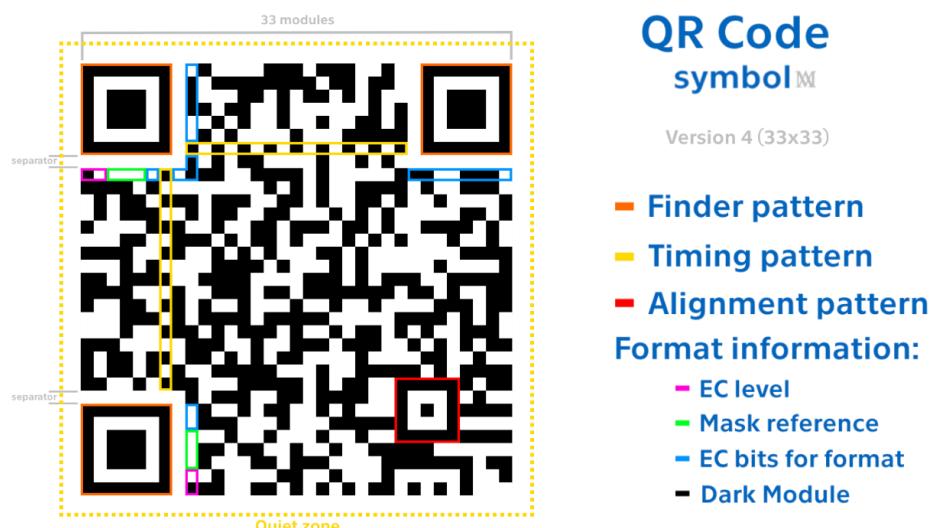


Figure 35: QR Code symbol. Based on [33][56]

Finder patterns are the most distinctive feature of QR Codes. Three "cornerstones" serve for detecting and locating the position and correct orientation of the symbol during reading. They are separated by blank gaps (separators) from the rest of the symbol. Timing patterns connect all 3 cornerstones together with a B&W alternating line. These help to determine the version of the symbol and help with coordinating the module's position. Alignment is an additional element of Version 2 and greater. For bigger symbols, there are multiple Alignment patterns arranged in a grid. They

ensure correct reading despite bulging and other distortions of the symbol. [33][56]

Format information is doubled string of 15 modules, containing EC level reference in the first 2 bits, followed by mask pattern reference in the next 3 bits and the rest are EC bits just for Format information. This information is doubled due to its essentiality for decoding the symbol. In the remaining area, data are encoded in 8-bit (4x2 modules) CWs along with EC CWs. The layout pattern of CWs goes from the bottom right corner up and down from right to left. The first CW is 4-bit and symbolizes encoding mode (Numeric, Alphanumeric, Byte, Kanji, etc...), after data and EC CWs follow. [33][56]

Error correction enables to store high amounts of data reliably. QR Codes use RS EC optional in 4 different levels. Naturally, the higher the EC level is the lower capacity for data. The ability to recover some amount of data enables to cover parts of the symbol, for example with logos, to make the symbol more receptive. Table 1 shows different levels of EC for QR Codes. [33][56]

<b>EC level</b>	<b>Recovery capacity</b>	<b>Binary indicator</b>
L (Low)	7%	01
M (Medium)	15%	00
Q (Quartile)	25%	11
H (High)	30%	10

Table 1: QR Code EC levels [56]

Masks or masking patterns break the clusters of same-colored bits that may appear after encoding the data. It is preferable for the modules to be arranged in balanced proportions of B&W throughout the data area to ensure reliable reading. One of 8 masks (see Attachment 9) is selected based on the penalty score (ability to eliminate the clusters) and applied. Demasking of QR Code from Figure 35 is depicted in Figure 36. Light grey bits are to be inverted from white to black- and chocolate-colored bits from black to white in order to demask the symbol. [56]

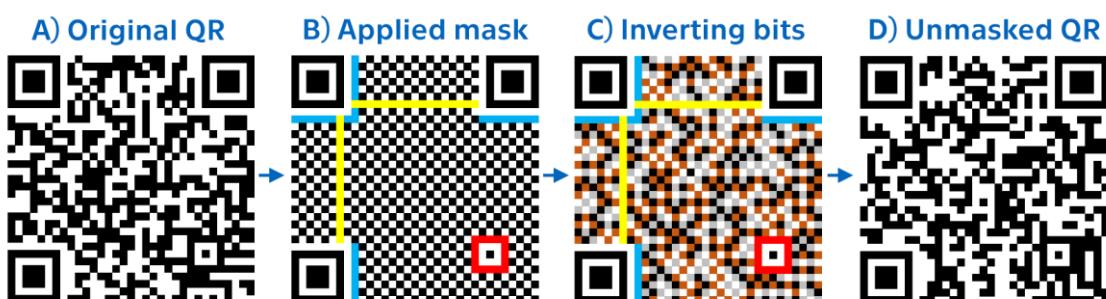


Figure 36: QR Code demasking. Based on [56]

Encoding and decoding of QR Codes include a set of complex calculations. But for contemporary mobile devices, such operations are completed within fractions of a second. The process of reading the QR code is to be seen in the UML Activity Diagram notation in Attachment 10. Outside its high capacity and quick readability, QR Codes have other advantageous features similar to DM such as color inversion and structured append connecting multiple symbols together. Modules or pixels can also be round, which is advantageous for direct part laser marking. Though, originally designed for part manufacturing, today, QR codes have become frequent for authentication, consumer advertising, payments, government applications, and many other. [33][36][41][56]

### 3.4 2D Codes for AR

Advancements in camera, computing, and SW technologies, allowed Computer Vision (CV) field to thrive. Using methods for acquiring, processing, analyzing, and understanding digital images and even real-time videos, computer brains got their "own eyes" to process data from the real world. Hand in hand with this field goes visual Augmented Reality (AR), which has become a subject of interest for both, industry, and academic research. An AR generates a composite view of the real world enriched with additional multimedia information, that serves to provide users with an enhanced interactive multi-sensorial experience. [33][57]

Despite looking like smaller versions of previously mentioned 2D codes, a symbol suitable for AR applications is usually called Fiducial Marker (FM). These serve as a reference point in 3D real world for a computer to determine the distance and orientation of the object FMs are attached to. The CV recognition of FMs is performed by analyzing contour structures with high contrasts (B&W) and template matching to the library fingerprint. Usual 2D codes carry some recognition data (orientation and finding patterns, mode data, etc.) plus a lot of information data, thus are very computation heavy. On the contrary, FMs carry mainly recognition data for fast and easy readability and only a small amount of information data, usually only the Identification Number (ID) of a marker. [33][57]

Following sub-Chapters are introducing 3 different libraries of AR markers. Examples of each library symbols are in Figure 37.



Figure 37: AR marker symbols [57][58]

### 3.4.1 ARToolKit

Developed by Hirokazu Kato in 1999, ARToolKit is a pioneer open-source library used in CV for tracking or AR applications. The symbol of this library consists of a thick black square border with a white field in the middle filled with customizable eccentrically placed patterns. This library is old and became obsolete because it has a high computational cost compared to newer symbols. [57][58]

### 3.4.2 ARTag

This library is a progress compared to its above-mentioned predecessor. ARTag symbol consists of a 10x10 matrix with 2 pixels thick border, resulting in a 6x6 internal area consisting of 36 bits, 10 of which are dedicated to storing ID and the rest for EC and checksum. This robustness allows ARTag recognize partially occluded markers. It has also a far quicker matching process than ARToolKit. ARTag library can generate up to 2002 unique symbols. [57][58]

### 3.4.3 ArUco

ArUco markers are fast, robust, and simple FMs. Its symbol is a 7x7 matrix with 1-pixel thick border, thus internal data area is a 5x5 matrix. As an EC, ArUco employs modified Hamming code and can generate up to 1024 different symbols. Symbols are designed with a mathematical algorithm that minimizes the chance of misidentifying markers even if a few bits are not recognized. The algorithm also avoids generating markers that are too close to all black or all white, to avoid confusion for other objects in the image. ArUco library was integrated into OpenCV library, which is a widely used open-source cross-platform library of programming functions mainly aimed at real-time CV. Due to their reliability and universality, ArUco markers are one of the most used FMs among AR developers. [57][59]

## 3.5 Utilization of 2D Codes

How and where are we going to use what type of 2D code depends on its properties. Capacity and omnidirectional readability are the main reasons for 2D code popularity in many sectors. Table 2 shows the Alphanumeric and Numeric capacity comparison of selected 2D codes. Based on these 2 metrics and the vastness of their utilization, 3 barcodes emerge as the winners: Aztec Code, Data Matrix, and QR Code. Their universal utilization was enabled by improved cameras in mobile devices and increasing speed of internet data transfer. [33][36][62]

2D code	Alphanumeric	Numeric
PDF 417	1 850	2 710
MaxiCode	93	138
AztecCode	3 067	3 832
DataMatrix	2 335	3 116
QR Code	4 296	7 089

Table 2: 2D codes capacity comparison [33][36][50]

Compared to 1D codes, which can in terms of reasonable dimensions serve only as an index to the backend database, 2D codes can contain semantically comprehensive data and serve as a database themselves. [33]

I allowed myself to divide the utilization of 2D codes into 3 main groups, which are introduced in the following sub-Chapters:

### 3.5.1 Payments & Authentication

COVID-19 pandemic catalyzed the use of contactless and cashless payment methods and their use by the wide public. But even a long time before these unhappy events, a lot of banking apps offered possibilities to scan or generate QR codes which would automatically populate payee or payer account information, paid amount, etc. Other methods would generate a symbol with your credit card information that would be scanned by the cash register during checkout without the need of physically having to have the card. The most common methods for POS payments are processed through NFC technology (Google Pay or Apple Pay) accepted by all modern card terminals. NFC technology will be briefly introduced in Chapter 4.

Symbols of 2D codes are also used in authentication, logins, and verifications. You may have ever connected to password-protected WiFi by scanning QR Code. Google and Microsoft authenticator apps also use disposable QR

Codes and asymmetric cryptography for pairing new devices. Certain versions of 2D codes like HCCB or color variations of QR Codes can be used to store facial biometric data. [33][41][60][61][65]

### 3.5.2 Marketing & Advertising

Probably the most common encounter of an ordinary person with a 2D code is a QR Code on some leaflet, poster, or package of some product. These codes usually contain an external link with a URL (Uniform Resource Locator), colloquially termed a web address. The person who scanned the symbol can easily access any website without transcribing a complicated string of text. Such external links can originate from Static or Dynamic symbols: [62][63]

**Static symbols:** The information encoded is fixed and cannot be changed. Usually contains specific text or permanently encoded link of the end address. Since it must contain a full URL or message text, it tends to contain more data and thus be less compact. On the other hand, no one can manipulate the end address and free online generators are widely available online to create any symbols effortlessly.

**Dynamic symbols:** Are far more suitable for marketing purposes, usually performed via third-party providers. The symbol contains shortened URL to the provider website and database, which is assigned to the end address. This operation does not take any longer for a human to notice, but thanks to this extra step, any access to the end address via the URL redirecting can be accounted to a particular symbol. This also allows the collection of data about the people who scanned the symbol. The date and time, location of access, or even the device type from which they accessed the link. And since the symbols contain only the shortened URL, they can be more compact or made more visually appealing by adding logos and changing colors. And finally, the URL redirecting end address can be changed or modified. This makes Dynamic symbols very powerful in marketing. Nevertheless, such provider services are usually paid and control over the symbol can be lost. Figure 38 shows the difference between static and dynamic symbols.

External links can lead to informative and educative articles or try to sell us more products by engaging us in interactive campaigns. Business cards can have imprinted symbols containing contact information or links to social networks or even AR elements. Entertainment could also be included in this group and often makes use of AR. Possibilities of use are limitless, but the

primary function is to connect us with one swipe of a finger to desired information or destination. [33][36][41][63]

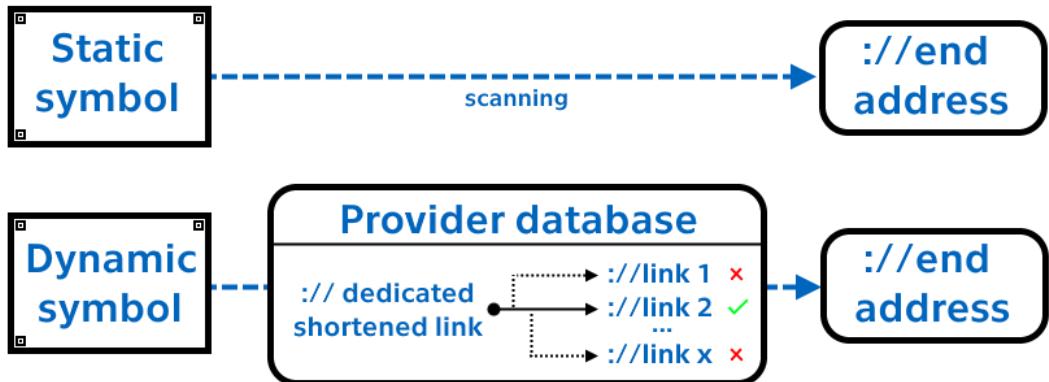


Figure 38: Static and Dynamic symbols. Based on [63]

### 3.5.3 Manufacturing & Logistics

Most modern products have been labeled with barcodes somewhere in their life cycle. For POS in retail, 1D EAN barcodes are still prevalent. But for transferring information between production steps, physical objects in logistics, or humans in personal transportation 2D codes are a perfect fit. Thanks to higher data capacity, they are suitable for labeling packages. GS1 barcodes also use standardized element string, which includes GTIN, batch or lot number, production, best before, and expiration date in format YYMMDD, or serial number. [33][41][64]

High-speed scanning capability is put to use in scanning moving parts, products, and packages on assembly lines or conveyor belts. Tracking, warehousing, or management of inventory is far more effective when using 2D codes. More data can be accessed by the BR without the need to go look up additional data in a database. Interconnectivity with ERP and other enterprise IS might provide a high level of data traceability. [33][36][41]

A very interesting trend with great potential is AR use in the industry. Utilizing devices like AR glasses or just regular smartphones with a camera, AR can serve during the training of new employees to perform assembly steps correctly, during quality control, for facility monitoring, various robotic applications, and also during assembly and maintenance. One such example is SW tool called EcoStruxure Augmented Operator Advisor developed by Schneider Electric to monitor and help with the maintenance of its automated assembly lines using only a mobile device with a camera. [66][67]

## 4 Alternative Technologies

Optical 2D codes are not the only technology that is used for AIDC. In this Chapter, I am introducing other AIDC technologies that are used in similar applications as 2D codes and thus are the main technological competitors.

### 4.1 OCR

Optical Character Recognition (OCR) belongs to the group of optical AIDC technologies. OCR systems can convert printed human-readable text into machine-readable text by utilizing a combination of HW for scanning and SW for converting the image into a B&W bitmap to distinguish the light areas as a background and the dark areas as characters. SW further processes the characters to find alphabetic letters or numeric digits, targeting one character at a time. Modern OCR systems can also make use of Artificial Intelligence (AI) to implement more advanced methods, like identifying languages or styles of handwriting. [68]

This technology is used since the 1990s while digitizing historical newspapers and today serves in many business applications like data entry and mining, automated accounting, and document processing. OCR accelerates workflows and reduces costs but is not suitable for outside the office environments due to slower scanning and heavier processing. [68]

### 4.2 Magnetic Stripes

Developed by IBM company for US government security purposes in 1969, the magnetic stripe technology in combination with POS terminals introduced by Visa became prevalent payment card technology for several decades. Information is encoded onto a ferromagnetic stripe containing tiny magnets oriented in a specific pattern, facing the north magnetic pole either up or down. Data from the strip are read by a solenoid that induces an electrical current in different directions based on the pattern in the strip as the card with the stripe is swiped through the reader. Because of data security issues and inconvenience during the payments, magstripe technology became obsolete and the Mastercard payments network is going to be the first to phase it out. [69][70][71]

## 4.3 IC Chips

An Integrated Circuit (IC) or Chip is a set of electronic circuits consisting of components like transistors, resistors, or capacitors designed to perform different functions within electronic devices. Since its invention, ICs are getting smaller and more compact. Thanks to that, ICs substituted magnetic stripes, because they can store and transfer far more data and perform various calculations. But to perform the desired function, ICs need electrical power to operate. A distinctive pattern of metal contacts provides the power to the IC from the reader. Such a pattern is used on SIM cards or payment cards that have embedded ICs inside. Chip payment cards, offer far more data capacity and are difficult to counterfeit and decrypt compared to magstripes, but require physical contact to transfer the data. [69][72][73]

## 4.4 Radio Frequency Technologies

### 4.4.1 RFID

Radio Frequency Identification (RFID) is a technology that uses radio waves to identify a tagged object. This wireless AIDC system consists of readers with antennas that emit radio waves and receive signals and tags that use radio waves to communicate and transfer information to nearby readers.

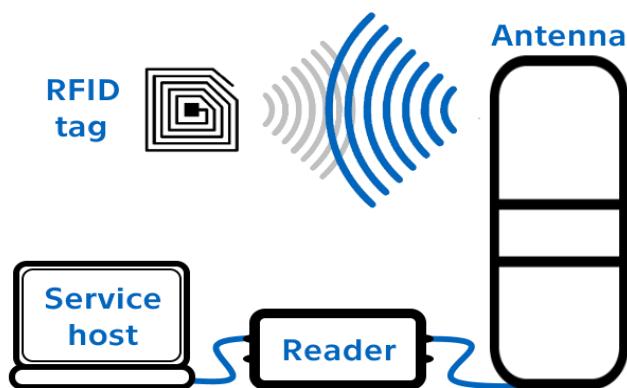


Figure 39: RFID system. Based on [74]

Tags can be passive or active. Passive RFID tags are powered by the reader through radio waves and do not have a battery. Active RFID tags are powered by batteries that enable bigger distances for reading but come at a higher price. Tags are made of an IC chip (small microprocessor with up to 50 thousand transistors), a coil antenna (that transmits, reflects, and collect power for the IC), and a substrate that holds the components together and

protects them. The RFID system is depicted in Figure 39 and an example of RFID gate antenna with visible coils is in Attachment 11. [74][75][76]

For RFID transmissions 3 frequency ranges are used for different applications: Low Frequency (10 cm), High Frequency (1 m), and Ultra-High Frequency (15 m). With active tags, the reading distances can be tens of meters. European countries have standards for communicating via radio waves and thus only a limited frequency range is allocated to RFID channels. Metal and liquids cause interference for RFID applications, but they can be mitigated with specialized equipment. RFID is mostly used for tracking and tracing objects, goods, pets, vehicles, or for access control. A very common application is a library book management system. [74][75][76]

#### **4.4.2 NFC**

Near Field Communication (NFC) is a subset of High-Frequency RFID technology. Similarly, it uses electromagnetic inductive coupling between two nearby loop antennas, which allows two devices to communicate when in close proximity. NFC devices can act as either a tag or reader. Secure bidirectional data transfer capability makes NFC perfect for access control and payment applications. Since the year 2010 NFC chips are used in smartphones enabling contactless payments using mobile wallets such as Apple Pay or Google Pay. NFC chips play a major role in post-sale marketing to enhance the product by linking relevant information, rating options, and reordering accessed only by scanning the product with your smartphone. Furthermore, to confirm the legitimacy of premium goods or add interactive elements. [33][77]

*NOTE: Your contactless credit or debit card most likely includes all: NFC chip, IC with metal contact plate and even the magnetic stripe has remained, despite technological progress, to be able to use different payment terminals. [69]*

### **4.5 Comparison**

For industry process purposes contact technologies are naturally not ideal. Optical technologies require a clear line of sight to read the tag properly. Out of this group, OCR does not seem ideal since it is computationally demanding compared to 1D and 2D codes and has no EC. Radiofrequency technologies have a huge advantage of not requiring a clear line of sight and thus tags can be hidden. RFID systems can locate hundreds of tags per

second. This makes them perfect for inventory management purposes. But radio frequencies may be disturbed, and industrial environments tend to abound with disruptive material - metal. [33][74][77]

Common passive tag carries about 2 Kilobytes (KB) of data, but passive Ultra High-Frequency tags can store up to 8 KB of data, which surpasses even the biggest 2D codes. Another aspect is price. RFID and NFC tags can be obtained for as cheap as 0,08 EUR up to 20 EUR per tag, depending on the level of customization and specialization. Contrary, 2D code symbols can be printed or engraved for a negligible price, or even projected on displays, not having to be materialized. Readers for radiofrequency technologies need to be specialized. For barcodes, you can use a ubiquitous universal device – your smartphone. Barcodes are considered safer than RFID, despite the possibility to fit simple scripts into Version 40 QR code. [33][50][56][74][76][77][78]

Ecology is another important aspect. Despite the low price of RFID tags, it is a specialized component with the whole industry behind it. The same goes for the reader systems. The ability to use universal tools to implement 2D code technology into any process makes this technology eco-friendly compared to its competitors. The following Table 3 shows the comparison of two types of each technology: [33]

high	medium	low	Data accuracy	Data capacity	Reading distance	Reading capability	Human readability	Operation cost	Tag cost	Reader cost	Security	Bidirectionality	Ecology
1D code													
2D code													
RFID													
NFC													

Table 3: AIDC technologies comparison. Based on [33][43][...]

None of these technologies is generally better than the other one. Each has its most suitable specific utilization and thus complements the remaining ones. 1D barcodes, like EAN, will probably remain a universal retail POS standard. RFID is the best for inventory management or tracking, 2D codes for logistics, packaging, or marketing, and NFC for secure contactless payments. Most probably, all these technologies will make further progress but still complement one another. [33][74][77]

## 5 Used SW Tools

In this chapter, I am introducing SW tools that are further used in the practical part or were utilized during the creation process of this thesis. Apart from those closely mentioned in the following subchapters, I should also mention indispensable MS Office, image editor Paint.Net, and video editor Sony Vegas.



Figure 40: SW tools logos [79][81][82][83]

### 5.1 Python

Python is a computer programming language created by Guido van Rossum and first released in 1991. It can be classified as **high-level** (using strong abstraction), **object-oriented** or OOP (encapsulates code within class-based objects), **interpreted** (processed by the interpreter into machine-readable bytecode), **structured and functional**. Python is a **general-purpose** language, meaning it can be used to create a variety of different programs and can be easily integrated with other languages such as C, C++, Java, and many others.

Python is one of the most popular programming languages. Its highly readable design and simple syntax make it beginner-friendly. It often uses English keywords whereas other languages use punctuation. Popularity can also be caused by the lightness and wit of its creators. The name itself refers to the British comedy group Monty Python. Furthermore, the big international coding community shares and explains various problems and examples on internet forums. There is also a huge amount of free audio-visual Python-related educative content on the internet. The current version that I worked with is Python 3.10.1. The logo of Python is in Figure 40 A. [79][80]

#### 5.1.1 Python Enhancement Proposals

A Python Enhancement Proposals (PEP) are documents that describe new proposed features, like design and style, for and from the community. [79]

**PEP 8 - Style Guide for Python Code** describes conventions, guidelines, and best practices on how to write Python code.

**PEP 20 - The Zen of Python** states guiding principles for Python's design using simple aphorisms. A few selected examples are:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Sparse is better than dense.
- Readability counts.

### 5.1.2 Syntax

The syntax is a set of rules that defines how a program will be written and interpreted. Python uses vertical text indentation to represent the block of the code, rather than curly brackets and semicolons as other languages do. Blank lines are ignored by the interpreter but serve as a visual differentiation and further improve code readability. Basic elements of Python code are briefly described below: [79][80]

- **Variables** are created the moment a value is assigned to them. They can be global or local. Naming any identifier has its best practices but is limited only by not using special reserved keywords.
- **Data types** are the narrower classification of data items. For example, numeric data type includes integers (int) or a real number with floating-point (float). The text data type is a string of characters denoted with single ('str') or double ("str") quotation marks. Boolean data type (bool) can only be one of two possible values: True or False.
- **Collections** are containers for holding data. Lists (`l = ['a', 'b']`) are mutable and tuples (`t = ('a', 'b')`) immutable collections of data. Dictionaries store data in key-value pairs (`d = {'a':1, 'b':2}`).
- **Control statements** include indispensable conditions (if-elif-else), for and while loops, and many others such as try, with, break, return...
- **Functions** are defined (def) blocks of code packaged as a unit, that can perform a specific task when called. Methods are then functions associated with the specific class they belong to.

### 5.1.3 Libraries

Python Libraries are collections of useful functions, utility methods, classes, and modules that eliminate the need of writing the code from scratch. Python has a comprehensive in-build standard library but there are also thousands of third-party libraries that can be used to enhance the code functionality in different fields of interest. Below I am introducing those I am using further in the practical part: [79]

- **Pandas** stands for Python Data Analysis Library is primarily focused on data manipulation and analysis.
- **SQLite** is a library allowing to manage local SQL (Structured Query Language) databases by a variant of the SQL query language.
- **Tkinter** is the Graphical User Interface (GUI) framework built-in standard Python library, allowing to create of windows, buttons, etc.
- **Pillow** (Python Imaging Library) offers image processing capabilities.
- **PathLib, Shutil & OS** allow to access filesystem paths, manipulate files in those directories and interact with the operating system.
- **Datetime** can be imported to work with the date and time.
- **CSV** allows working with the Comma-Separated Values (CSV) format.
- **QRCode** offers to generate QR Code symbols with various parameters.
- **OpenCV** is a cross-platform, open-source, and very popular library vastly used in real-time computer vision and machine learning.

## 5.2 PyCharm IDE

Integrated Development Environment (IDE) is an SW tool supporting writing and testing the programming code in a single Graphical User Interface (GUI). There are many different IDEs but I used PyCharm, specially developed for Python. The software enables the creation of virtual environments which ensure correct dependencies of all used module versions for each project. PyCharm also checks the correctness of syntax or gives hints at the PEP 8 best practices and provides many more assistant tools. Visual customizability allows using different colors for different code elements, which further increases the readability of the code. Attachments 13 and 14 are showing examples of code described in Chapter 8 in this customized environment of PyCharm. Logo of PyCharm IDE is in Figure 40 B. [81]

### **5.3 GitHub**

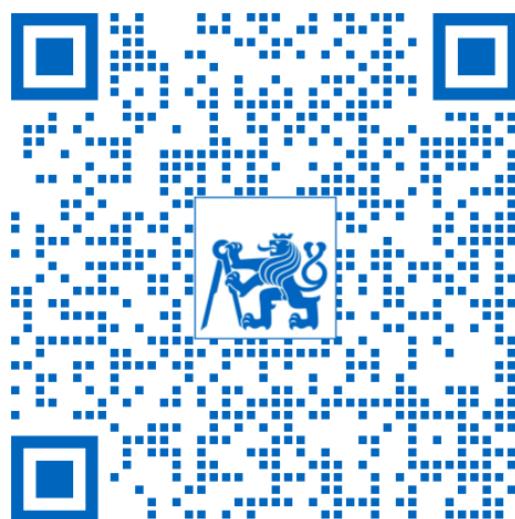
GitHub is a code hosting platform for version control and collaboration using Git SW. It allows programmers to track changes, coordinate group projects, and share or publish them online. It also serves for storing open-source code in online repositories, where other members of the community can access, debate, copy and enhance other people's code. I am using the online repository to store the code of my practical part project. The logo of GitHub is in Figure 40 C. [82]

### **5.4 Lucidchart**

Lucidchart is an online browser application that allows the creation of various types of diagrams and improves processes, systems, and organizational structures. I am using Lucidchart for diagramming the processes in upcoming Chapters in BPMN 2.0 and UML notations. The SW allows creating several diagrams with a limited number of objects for free. The logo of Lucidchart is in Figure 40 D. [83]

## 6 APS Systems

APS Systems is a virtual tech company created for the purposes of the course Projekt 3 taught by Ing. Miroslav Žilka, Ph.D. at the Czech Technical University in Prague. Although it is a virtual company, its creation is based on thorough research and real data. The company is the result of the teamwork of Petra Brtková, Arkadii Makhmatov, and me. We designed a complex company structure and fabricated the whole context of existence for this company. The requirement for this course was to create an elaborate MS Excel file simulating various company processes and functions. We also had to compile the final report and presentation. As these documents serve as a source of information for the following Chapters, I publicly published them on the GitHub repository. You can access and preview them with the following [QR3](#). The documents are available in the Czech language only.



QR 3: APS Systems documentation [84]

I decided to use APS System company for my thesis as I had a deep insight into the company's functioning and could also imagine various difficulties that the company if ever have been existing, might have encountered. All the information that is directly obtained from the original work is cited with the [84] index. For the purposes of this thesis, I had to deepen the detail of some topics. Such information has no citation as it is solely my work.

## 6.1 Company Introduction

APS Systems s.r.o. is a medium-sized Czech tech company that develops, produces, and sells premium products for garden maintenance. The main product is a highly autonomous robot lawnmower. The company stands out from its competitors, with its product's premium quality and Hi-Tech functionalities. The company's efforts are to combine the latest technologies with environmental sustainability. The logo of the company and overview of basic information for the year 2020 are stated in Table 4 below: [84]

Name	APS Systems s.r.o.	
HQ	Technická 58, Unhošť 273 51, CZECHIA	
DIČ	CZ42069666	
Employees	86	
EBIT	CZK 139,631,932	

Table 4: APS System logo & information [84]

The vision of the company is:

„To develop intelligent systems for households with regards to safety and comfort of our customers.“

The motto of the company is:

„The best maintenance for your garden.“

The company was established in 2014 as a detachment of sister company APS Solutions, with the ambition to focus on high-tech robotic lawnmowers and in-house developed products. APS Solutions had existed since 2002 and successfully cooperated with German agriculture machinery manufacturers on SW development and has rich know-how in this field. APS Solutions is a main strategic partner and as such is an exclusive supplier for SW and participates in product development. Both companies also share offices.

In 2018, the product APS Meower™ (see Chapter 6.3) was introduced - a unique autonomous robotic lawnmower with a collection basket. The production facilities were upgraded, and manufacturing started. Despite its high cost, it became an immense selling success. [84]

In 2020 the plans for expansion to North American (NA) market are introduced and the company is challenged with increasing demand. In 2022 company grows in both numbers of personnel and satisfied customers.

## 6.2 Organizational Structure

In the year 2020, out of 86 company employees 67 worked in the Operations department, 9 in Finance, and 10 in Sales. This may seem like a very low number of personnel, but the company has well-designed processes and is supported by robust but easy-to-use IS and ERP that are ensuring maximal effectiveness of human work, so employees do not have to waste their time with administrative tasks. By 2022, the company grows in revenues, the volume of production, and the number of personnel. The organizational structure of the company is shown in Figure 41. [84]



Figure 41: APS Systems Organizational Structure [84]

## 6.3 Products

The main product and the flagship of the company is APS Meower™. It is a premium quality electro-robotic lawnmower. This unique solution not only mows the grass but also has the ability to collect mowed biomass while being completely autonomous. Compare to its competitors it is the only device that could be considered fully autonomous. This is because the machine is equipped with various sensors and a powerful control unit with modern navigation, security, and mapping systems. This high-tech solution ensures safety for the product itself, but also its environment. It can be controlled via APS App (desktop or smartphone), where scheduling and other parameters can be set. [84]

The machine (displayed in Figure 42) is powered by 4 electric in-wheel motors with independent rotation. This enables a 360 ° turn on the spot. The three cutting electrically driven drums have a total cutting width of 0.9 m. Biomass is discharged into the collection bin with a capacity of 80 liters. The machine can empty the bin at a designated location, which can be set in the APS App. Electro-hydraulic chassis allows adaptive cutting height in the range of 20–100 mm.

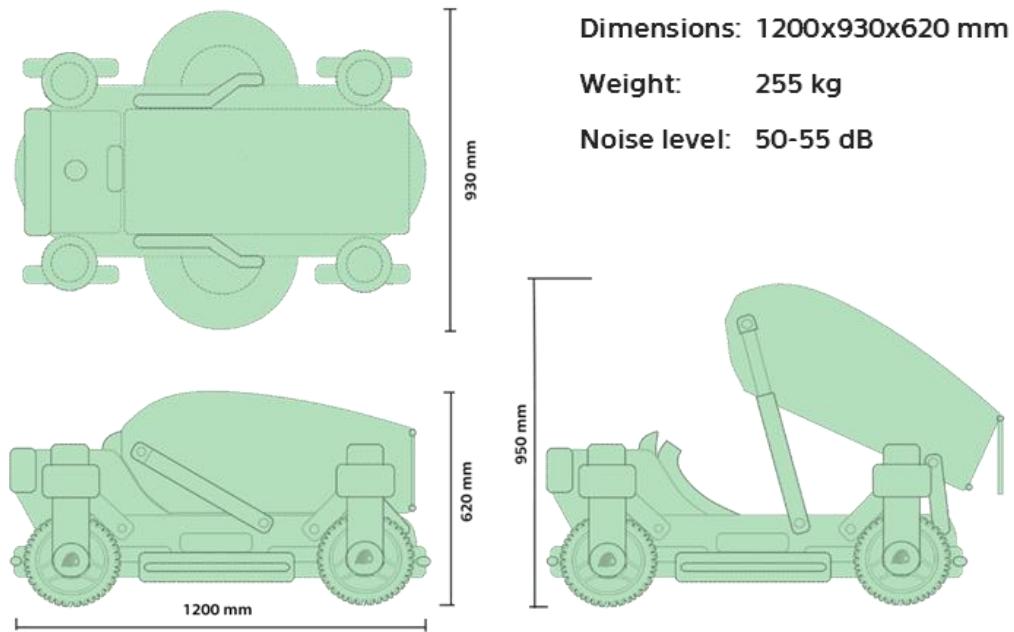


Figure 42: APS Meower [84]

The philosophy of the product is to take care of the client's garden independently with respect to the maximum safety and efficiency of the whole system. The product matches the latest trends in Smart Home connectivity and ecological biomass management, namely support for composting (natural processing of biowaste for further use). The machine can also operate in the mulching mode, the cut biomass particles are returned to the lawn, where it serves as natural fertilizer and protection from soil erosion. The APS Meower can also handle fallen leaves and small branches, which is a welcome help in the autumn season. [84]

Other products of the company are complementary to the main product (composter, SW extensions, winter snow-blowing kit). Generation 2 of APS Meower is currently in the R&D process.

## 6.4 Customers

As the company's product is a luxury item with a high price tag. The main focus of the company in the Business to Consumer (B2C) sector is mainly on households with above-average income and with a big garden or vast property. These groups might be interested in the uniqueness of the product as well as their own internal interests in ecology and the environment. In the Business to Business (B2B) sector, the potential customers are companies focused on the maintenance of public greenery, the maintenance of industrial grounds, or companies offering garden-related services. APS Systems serves their customers in cooperation with European distribution partners or in their own Show Room. This distribution network is displayed in Figure 43. As already mentioned, expanding the network to the NA market and potentially globally is in progress. [84]



Figure 43: APS System EU distribution network [84]

## 6.5 Suppliers

APS Systems operates on an automotive-like business model, so reliable suppliers are the key element. The company outsources the manufacturing of plastic and metal parts for the product as these components require to have special and expensive manufacturing equipment. A big part of the Czech Republic's industry is automotive. A lot of Czech companies are supplying various parts to the automotive industry as tier one or tier two suppliers. For this reason, it is easy for APS Systems to find a partner who wants to fill free production capacities with the company's smaller batches.

Parts such as electronics, sensors, and hydraulics are purchased as standardized components from wholesalers. Complex circuit boards and ICs are purchased as custom products, simpler ones are in-house produced. [84]

## 6.6 Production Process

A big portion of the inputting components and materials are purchased from or outsourced by suppliers. The company has its own Welding Shop (WS) where the metal chassis is welded, and Paint Shop (PS), where the chassis and other metal parts are coated with protective layers of paint. There are two types of an assembly lines. Assembly 2 (A2) is on the second floor of the production facility and here all the electronic components are soldered on circuit boards, connected together, and tested. Assembly 1 (A1) completes the final product in several steps of the assembly line. The final step is Quality Control (QC) station, where all safety and functional features are tested and checked. The whole manufacturing process is diagrammed in Figure 44. The production teams participating in the process are also visible in the facility layout in Chapter 7.1. [84]

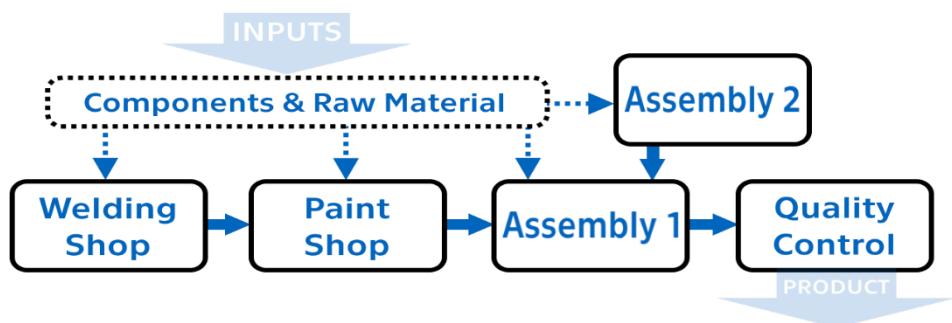


Figure 44: Manufacturing process of APS Meower [84]

## 7 Logistics & Inventory Problem

To understand what improvement is needed, this chapter describes the current state of the APS System's facility, internal logistics of supplied parts, and motivation to realize changes.

### 7.1 Current Situation

APS Systems has its own production facility, Headquarters (HQ), development center, and offices in one building. Figure 45 shows the floor plan of both levels of the building. The company shares some of the office's capacity with its sister company APS Solutions, to stimulate collaboration.

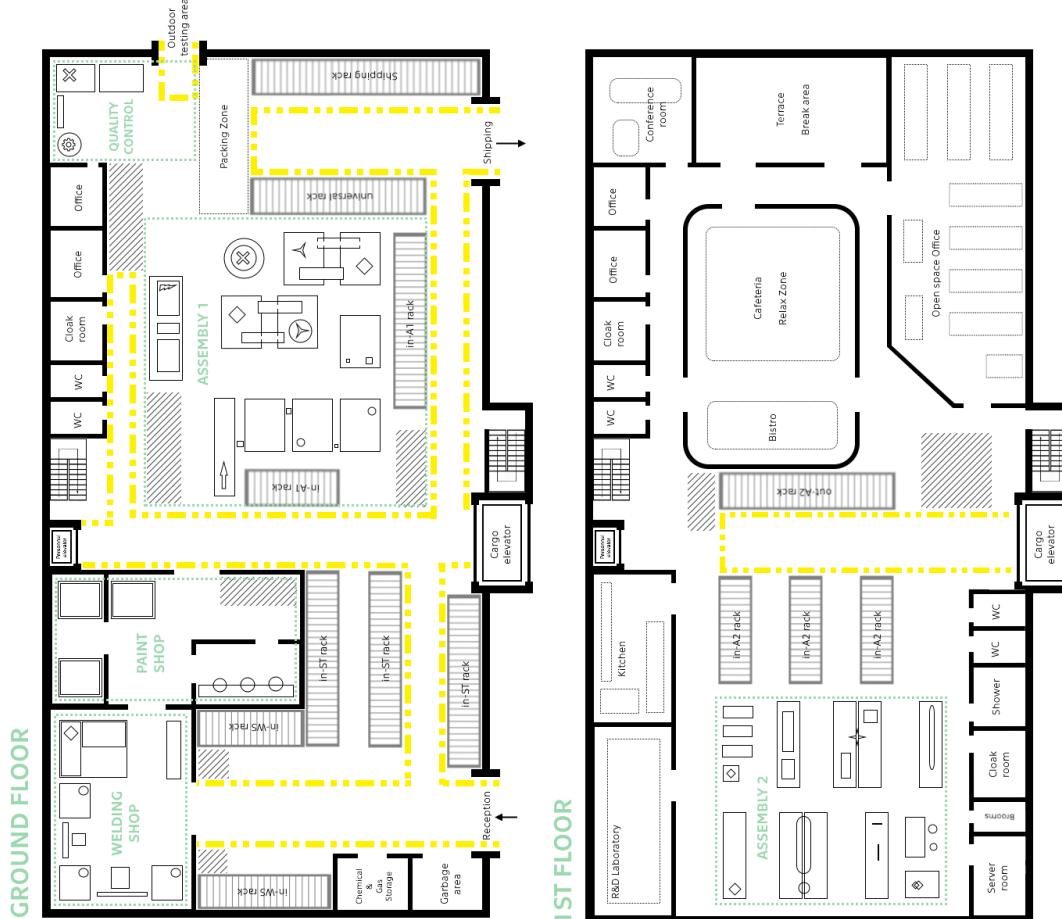


Figure 45: APS facility floor plan

As already mentioned, the company purchases and outsources the majority of components for its products. This comes with big requirements for logistics and warehousing capabilities. The company's suppliers are contractually bound to make regular deliveries. This comes from the need to

minimize storage space. Despite these efforts, some storage capacity is always needed. The facility (built in 2015) was designed in a way to have a source of components as close to the manufacturing or assembly area as possible. That is why the company has 5 different input Warehouse Storing Spots (WSS). These are the internal designation of racks or areas, where raw materials or supplied components are stored. As all WSS store **inputs** for production, they are labeled as: in-XX (using some of the abbreviations from Chapter 6.4) with respect to the production process flow. These labels (depicted in Figure 46) serve for quick and easy recognition and orientation.

- **in-WS** (input - Welding Shop) stores all the metal parts and raw materials used in WS for chassis manufacturing.
- **in-CG** (input - Chemicals & Gases) is a specially secured room where all the potentially hazardous substances are stored. These include welding gases for WS, paints for PS, and others.
- **in-A2** (input – Assembly 2) stores the inputs for the assembly of electronic units in A2, namely electronic components.
- **in-A1** (input - Assembly 1) stores components for final A1 assembly.
- **in-ST** (input - Stock) is the universal WSS, with the biggest capacity, where reserve stock for the in-A2 and in-A1 are stored alongside R&D and QC materials, office supplies, and other items needed for facility and company operations.



*Figure 46: WSS labels*

This system was possible to implement in 2 level facility only thanks to the cargo elevator and the fact that all the components entering the A2 part of the production process (as described in Chapter 6.4) are rather small electronic parts. Storing stock of input items close to the workplaces was evaluated as more efficient as opposed to providing frequent internal logistic operations from the one big main warehouse. What the lifecycle of such part is, describes next Chapter.

## 7.2 Inventory Flow

All the supplied parts and components are ordered by Purchasing Manager according to production plans, Financial Planning and Analysis (FP&A) reports, and items currently in stock concerning the usual delivery times of each supplier. All to ensure that the components are available for planned production (plus security reserve) at the right time.

Arriving packages with components are usually in boxes labeled with classic 1D barcodes (commonly used in transport and logistics as mentioned in Chapter 2) and enter the building by "receive entrance" to the warehouse area. Here they are required to be scanned by a special BR that is connected to the company's database and can match the barcodes index to the correct item. This scanning is required for every package as it registers incoming item data in the company's IS. After takeover and scanning all the items, packages need to be sorted by warehouse staff either by reading text on the labels or scanning the barcode with the BR again in reading mode to access the desired storing location. Each batch of packages is then transported on the pallet to the respective WSS and stored in a rack. Figure 47 shows highlighted areas of each WSS, the logistic movement of input components, and Work in Progress (WIP) between WSS and production teams:

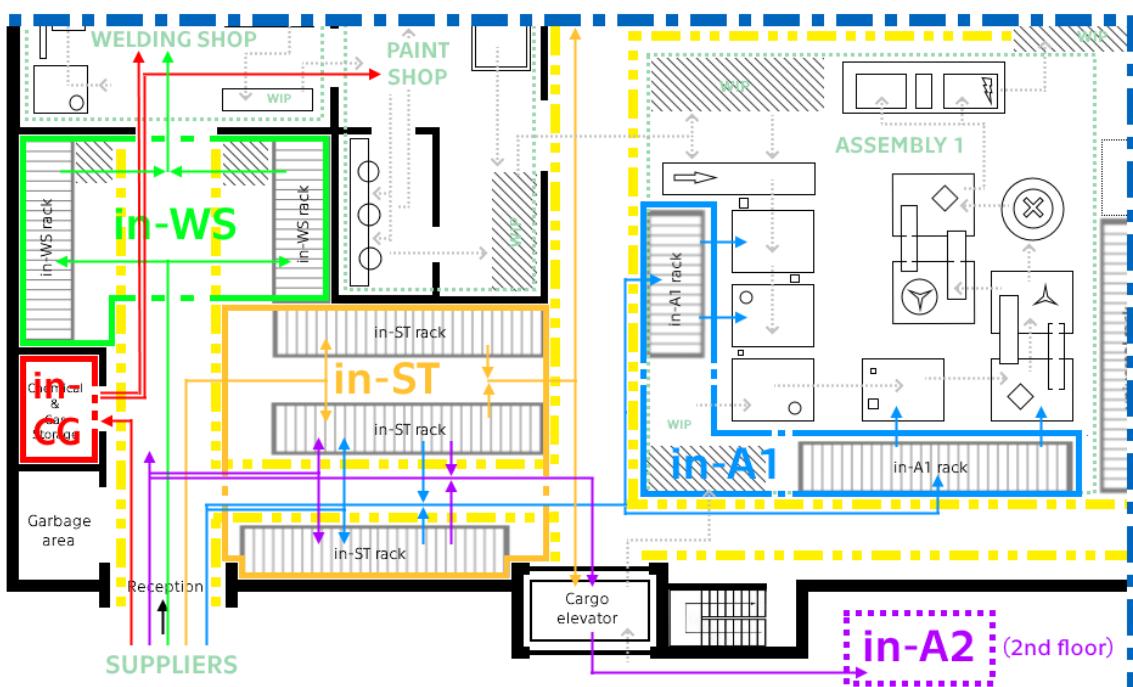


Figure 47: Inventory flow schematics in floorplan detail

Outside the input WSSs, the company has special areas for WIP to queue in between production steps. After the product is finished, it goes to QC, where it is thoroughly tested also in outside conditions and after it goes to the packing area to be cleaned, packaged, and stored in the output rack waiting to be shipped to distribution partners and customers.

### 7.3 Motivation

So far, this system is operating without major problems. That does not mean it has no weak points. There are several of them. If more deliveries come at the same time, the logistic staff is overwhelmed with packages that need to be registered and put in the correct place. Such a situation can be stressful for some people and thus increasing the possibility of human error.

The current solution using conventional BR for scanning and registering the incoming goods can be time-consuming and require focus not to omit any package. The problem also is that 2 steps need to be performed (sorting and scanning). If the packages do not contain a label with human-readable text, an operator needs to scan the barcode with BR connected to the company's IS to look up the WSS of the package. This also requires a wireless connection and if part of the wireless system does not work at the moment, the operator has no way to find out what is inside, apart from deducting from the delivery list or opening each box in the packages to see the content.

Another issue is that the packages are registered at one place, then transported and stocked to the racks in respective WSSs. But if the package is sorted incorrectly or disappears during the internal transport, the IS's database will have no updates about this. This process is very prone to human error and can cause painful issues.

Even if everything goes well and the item is sorted, transported, and stored as it should be, the IS has no detailed information about the exact location of the packages in racks within each WSS. Workers rely on the customary storing locations for different parts. But if a new worker does not know these customs or the desired rack shelf is already occupied, the item is placed elsewhere. Current IS is not able to reflect these deviations. Searching for parts wastes precious workers' time, but also since humans are creatures of habit, such inconvenience can worsen workers' mood and therefore negatively affect working morale.

## 8 Logistics & Inventory Solution

To improve the logistic process described in the previous Chapter it was decided to come up with a replacement for the current solution and redesign the package ordering and receiving processes. I also had to reflect on several criteria and requirements that APS System might have for the new solution.

- The sorting subprocess should speed up and be more faultless to prevent blockage in the receiving area.
- The registering should specify the exact location in certain WSS and should be done during the physical placement of the package to that location to prevent any misplacements and loss of packages.
- The solution should not notably burden the company's suppliers.
- Package labels should contain the content data, readable offline.
- Be cost-effective and easy for implementation and training.

There are many possible solutions to this problem. But taking into account all the knowledge from Chapter 4.5 and reflecting the company's requirements, I excluded the following options for respective reasons.

The RFID technology burdens the suppliers with the necessity of implementing tagging technology in their packaging process. Also, the company itself would need to invest in reading technology. Even though the mass reading process using RFID can be lightning-fast, this system variation does not suit the process where the packages should be matched to their storing location in contrast to retail clothes shops where one powerful reader can scan all the products in the shop within a few seconds. Individual scanning by a human-operated reader could solve this issue but does not speed up the process significantly compared to optical scanning. For these reasons, RFID is not viable, and therefore NFC is an even less relevant option.

Regular 1D codes are already employed in the process but are not able to hold all required data about the package and are used only for indexing. Even the addition of the text to the label does not seem to improve the process at all, thus 1D codes are not a suitable solution as well.

2D codes can be utilized using the same technology that is already in use. They offer big data capacity for offline package details reading. However, employing some 2D code will not speed up the sorting process, despite the possibility of reading either with the BR or personal smartphone.

## 8.1 Solution Proposal

For new labels used on packages, I proposed utilizing QR Codes as they have big data capacity and are quickly readable. But not regular QR Codes. Inspired by marketing purposed QR Codes, which have various logos imprinted on them to quickly recognize if the link encoded in the data field leads to a Youtube video, Instagram profile, or Company's website, I decided to use similar labels. To the QR Code containing data about the ordered package the WSS label (see Chapter 7.1) was added. This enhances the QR Code functionality by the addition of human-readable information.

To create such QR Codes, input information had to be prepared. The company had its own database of purchased parts that are repeatedly ordered. I modified the structure of the database to fit these new needs. The list representing the content of this database is in Attachment 12. The inside data structure consists of:

- Stock Keeping Unit (SKU) is a unique internal code consisting of a combination of letters and numbers that serves for the identification and tracking of inventory items within the company. [85]
- Name of the part
- Category identifier
- Warehouse Storing Spot identifier
- Supplier Lot Number is a number from the catalog of our suppliers

Most of these data are used in generating the label. My proposal also includes the Order ID which provides uniqueness and is needed to ensure traceability. I designed the Order ID format as shown in Figure 48, where YYYY represents a year, MM month, and DD day of the date when the order was created. For cases when more orders are generated in one day, X then stands for the sequence number of the order made that day, starting with 1.

**APS-YYYY-MM-DD-OR-X**

Figure 48: Order ID structure

The example illustrated in Figure 49 shows data from Supplied Parts Database, Order ID, and ordered quantity serving as inputs. The generated

output is then the standard QR Code symbol with a WSS label pasted in the middle, scannable by any device able to read QR Codes.

## INPUTS:

Database:	
<b>SKU</b>	→ QR Data
<b>Name</b>	→ QR Data
<b>Category</b>	→ <b>x</b> (Not used)
<b>WSS</b>	→ Label
<b>Lot #</b>	→ QR Data
Generated:	
<b>Order ID</b>	→ QR Data
<b>Quantity</b>	→ QR Data

## OUTPUT:

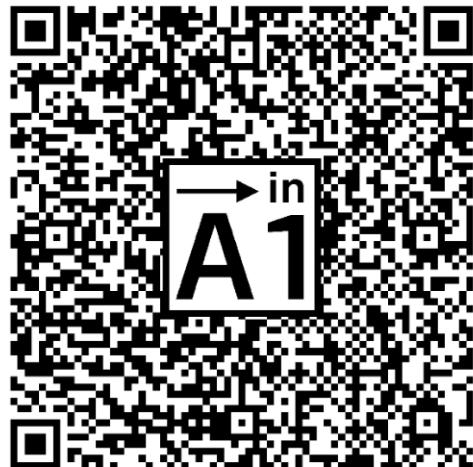


Figure 49: Encoding order data into QR label

## 8.2 New Process Workflow

To implement this proposed solution, an upgraded process of the whole logistic cycle of supplied parts (from the need of such part, until it enters the production process) needs to be designed. Participants of the process are internal (Planning, Purchasing, Logistics, and Production departments) but also external (suppliers and contractors). As part of the design of the modified process, it was necessary to create process maps or flow diagrams, which were introduced, including their usefulness, in Chapter 1.3. For modeling this process I am using BPMN 2.0 notation and the tool Lucidchart. The final design of the new process is depicted in Figure 50 on page 76. This diagram is described in detail in the following text with all the links to supporting tools utilizing QR Codes (introduced further in this thesis). The diagram will also serve to increase the exactness and precision, and clarify the division of responsibilities of all process participants.

The new process starts when the production and sales plans for the new time period are made based on the internal data, historic records, market research, and demand prediction models. Purchasing Manager receives this information along with the production schedule. His or her responsibility is to continuously check on warehouse stock levels and compile an ordering plan respecting the production volume, supplier delivery times, supplier

contract terms and conditions, and other factors. Each ordering instance should follow the diagram in Figure 50 on the next page.

If the specific unavailable components need to be ordered, Purchasing Manager generates a new order. This task has a sub-process that is a subject of my SW solution described in upcoming Chapter 8.3.1. As outputs of this task, a list of ordered items and previously described QR Code labels for each package are generated. These documents are sent to the supplier with the formal order protocol. The Logistics team is informed about the future delivery and is forwarded the list of ordered items for a future delivery check. For now, the job is done for the Purchasing Manager for this specific order.

The supplier receives order information and starts its internal process. The only new addition is the obligation to correctly label the packages that are about to be shipped to APS Systems. After shipping, the supplier's internal process is over, and the agreed payment can be expected if the shipment complies with the order.

Subsequently, APS Systems's Logistics team receives the expected shipment. If it would happen not to comply with the order, the Purchasing Manager is informed and triggers the Corrective and Preventive Action (CAPA) which is specified in the contract with the supplier.

Upon receiving the number of packages is checked to comply with the order list. Then the Logistics staff will sort the packages based on WSS destination. The new human-readable label within the QR Code allows doing this without any other tools. This task has a parallel label (3 vertical stripes) in the diagram as the sorting of potentially all of 5 WSS piles takes place at the same time.

The next task is stocking packages into each individual WSS. This is a sequential task (3 horizontal stripes) as the Logistics staff can only stock one package at a time. The stocking also includes another sub-process that is also a subject of my SW solution described in upcoming Chapter 8.3.2. During the realization of these tasks, the warehouse database is updated as the packages are scanned to their respective WSS, using my SW tool. Contrary to the original process, this ensures data credibility.

Upon finishing the Logistics tasks, the Production department is informed via Purchasing Manager, that the desired components are stocked and ready for production. Whenever a new order is created, the process repeats.

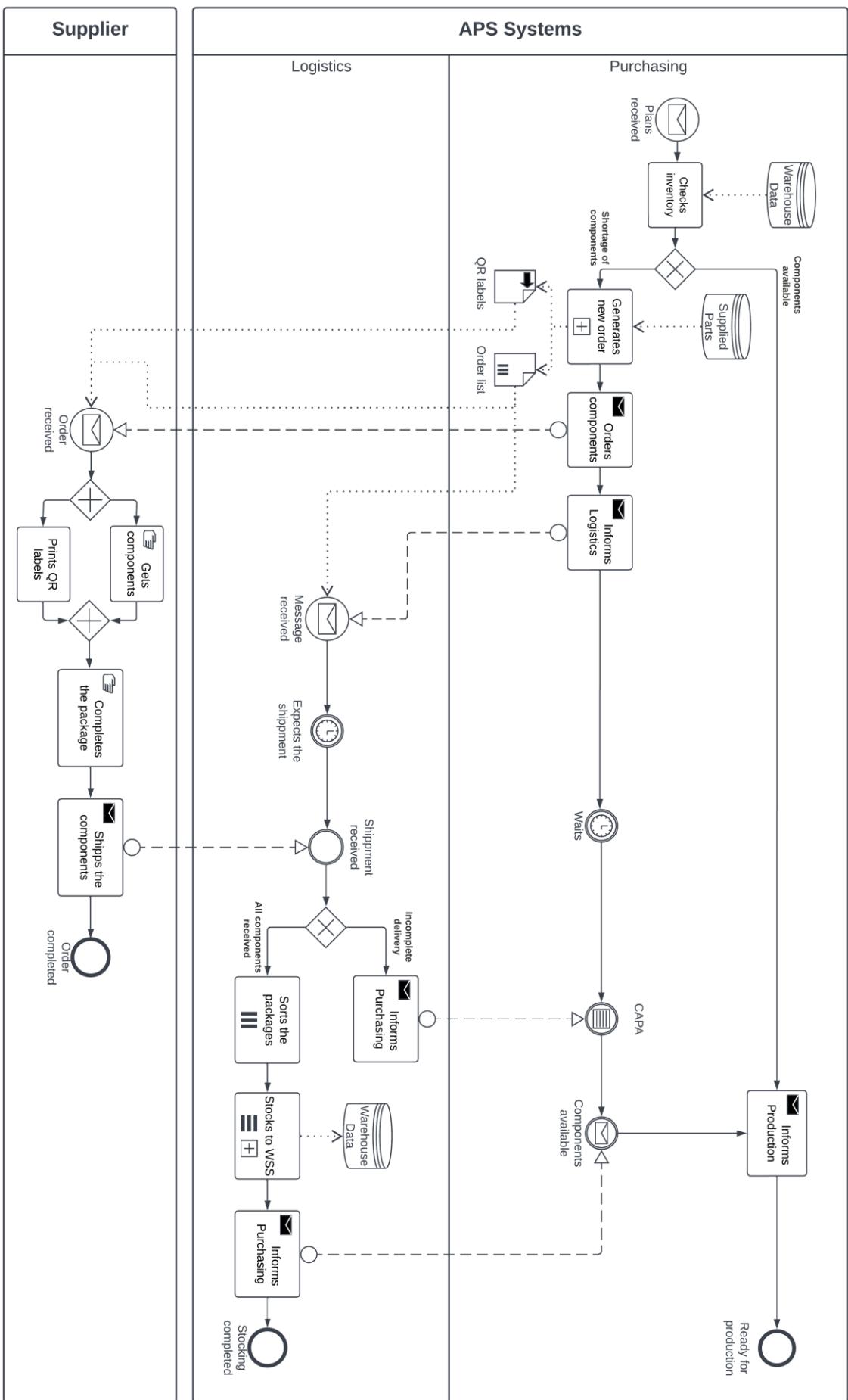


Figure 50: APS supply chain process in BPMN 2.0 notation

### 8.3 Python Application

As a part of the solution to the discussed problem, I set myself a goal to create SW tool, that will help smoothen the flow of previously introduced process by utilizing 2D codes to transfer vital data for the enterprise. The outcome of my Python programming efforts is the APS System App. It is a program, that is meant to help realize the proposed process upgrade. The screenshot of the APS System App Menu is in Figure 51.

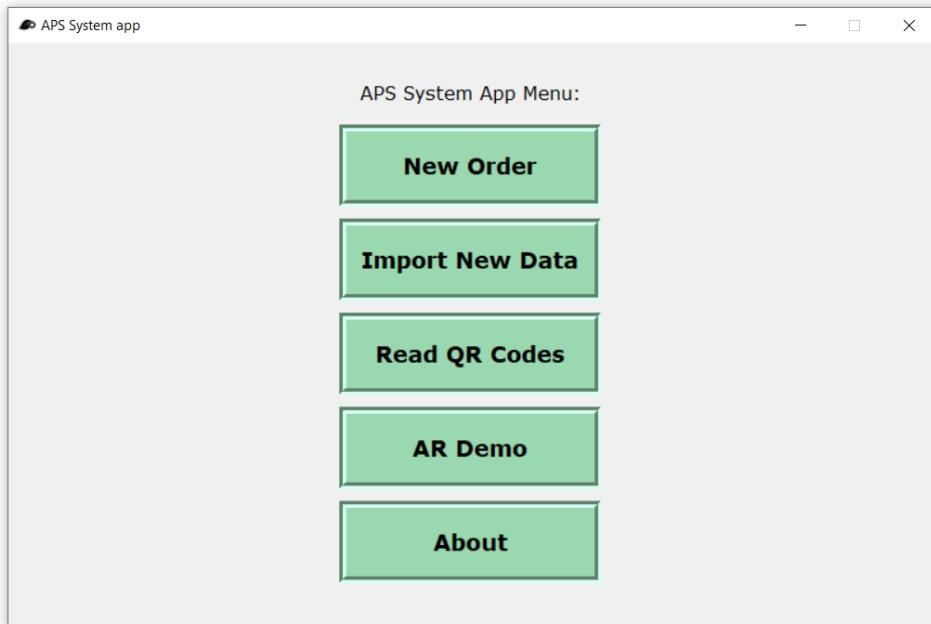


Figure 51: APS System App Menu window

The usage of a GUI environment is necessary as the final users – employees of the APS System, might not be how to control the program through the command line. The program has in total 6 GUI windows, closely described in sub-Chapter 8.4.2.

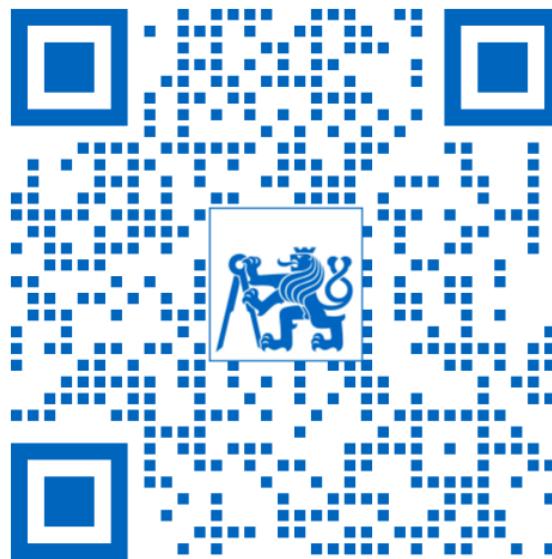
Running the Python program will open Window #0 (Figure 51) with 5 buttons. Each button leads to another window with a different purpose. The first button from the top marked "New Order" provides the option to generate a new order. It is the first sub-process mentioned in the previous Chapter and is thoroughly described in the following Chapter. The third button marked "Read QR Codes" is the other sub-process to be introduced in Chapter 8.3.2. These 2 represent the main functionalities of the program.

The second button – "Import New Data" serves rather as a support tool to the ordering system. The fourth button was intended as an AR utility

demonstration and this unrealized intention is closely introduced in Chapter 9.2. The last button shows the info about the program.

### 8.3.1 Creating New Order

Pressing the “New Order” button in the menu opens Window #1 (Figure 53) enabling the creation of a new order. The Youtube video on the following [QR4](#) link demonstrates the functionality of this feature. I recommend watching the video before further reading.



*QR 4: Video 1 – Creating New Order*

The window of the APS Item Ordering System has 2 inputs. The first is a drop-down menu where the desired component can be selected. The content of this menu comes from the internal database of supplied parts as the BPMN diagram indicated in Chapter 8.2. The source data for this database is appended in Attachment 12 and the sample is displayed in Figure 52 using the online browser SQLite Viewer able to read .db files. [92]

PARTS (51 rows)				
<pre>SELECT * FROM 'PARTS' LIMIT 0,30</pre> <span style="float: right;">Execute</span>				
SKU	Part_Name	Category	Storing	LOT
EL00021	InWheel electric motor	ELECTRO	in-A1	1143585376
EL00026	Stepper electric motor	ELECTRO	in-A2	8375540703
EL00015	Cutting electric motor	ELECTRO	in-A2	1707439044
EL00011	Control unit	ELECTRO	in-A2	1347542816
EL00032	Battery packs	ELECTRO	in-A2	9764407469
EL00124	Warning lights	ELECTRO	in-A2	3346675090
EL00123	Alarm buzzer	ELECTRO	in-A2	6327267085
EL00070	Wifi reciever & transmitter	ELECTRO	in-A2	5083061159

*Figure 52: Supplied Parts Database content*

After the item is selected, the user fills in the numeric value of the ordered quantity into the input field. By clicking the button "Add to the order list" an item is added to the order list and to the preview of this list in the middle of the window. If some item was added in error, it can be selected with a cursor and by pressing the "Remove selected from the order" removed from the order list. When the user selects all items that need to be ordered, pressing "Finish the order" will finalize and exports the order.

An example of an ordered list filled with 6 items is in Figure 53.

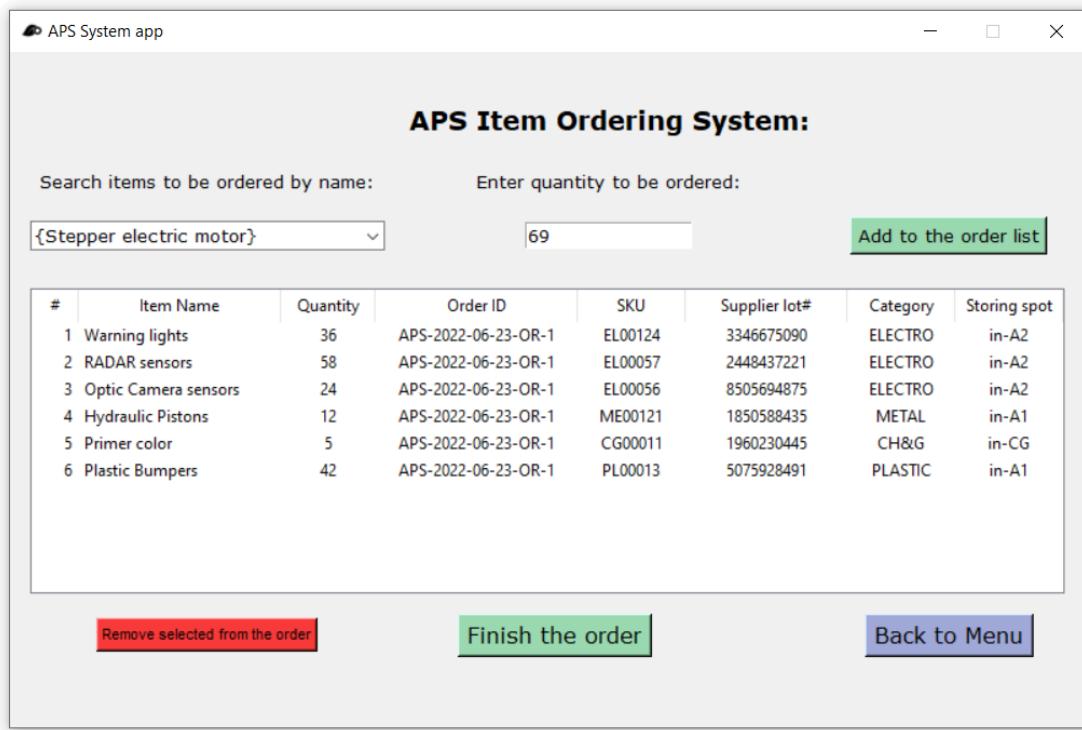
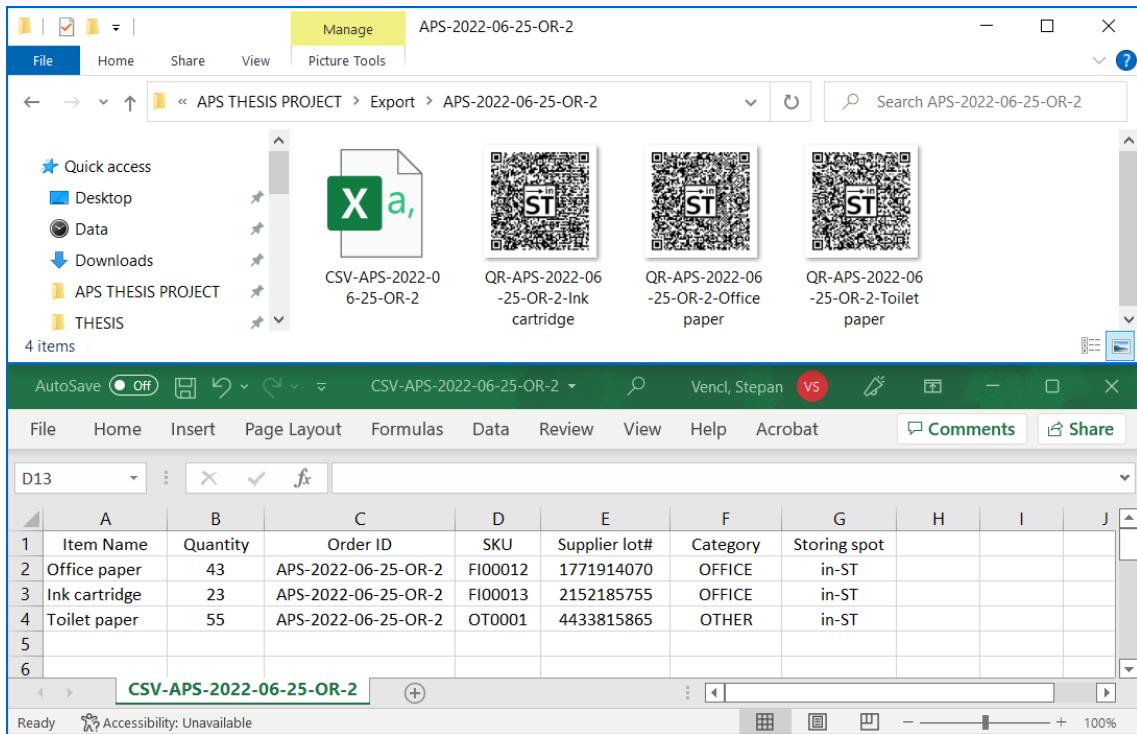


Figure 53: APS App Item Ordering System window

Finishing the order will create a new directory named after generated Order ID. In this directory, two types of outputs are automatically generated by the program. One is a CSV file with the order list info and the other are QR Code labels for each item in the list. Figure 54 is showing export directory content of the order for Order ID APS-2022-06-25-OR-2, that is consisting of 3 items in the upper part, and in the bottom part, the opened CSV file for the same order.

The labels have data structure as described in Chapter 8.1. These outputs should be sent to other participants in the process as indicated in the BPMN diagram in Figure 50.

The "Back to Menu" button is kindly taking the user back to the menu.



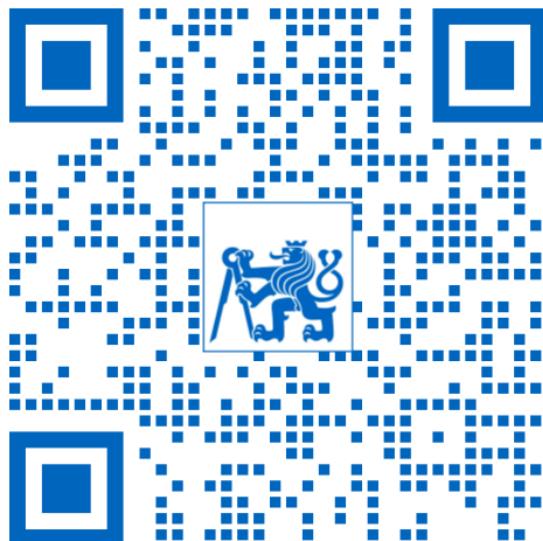
A screenshot of Microsoft Excel showing a spreadsheet titled "CSV-APS-2022-06-25-OR-2". The spreadsheet contains data for three items: Office paper, Ink cartridge, and Toilet paper. The columns represent Item Name, Quantity, Order ID, SKU, Supplier lot#, Category, and Storing spot. Below the spreadsheet, there are four QR codes labeled "QR-APS-2022-06-25-OR-2-Ink cartridge", "QR-APS-2022-06-25-OR-2-Office paper", and "QR-APS-2022-06-25-OR-2-Toilet paper".

	A	B	C	D	E	F	G	H	I	J
1	Item Name	Quantity	Order ID	SKU	Supplier lot#	Category	Storing spot			
2	Office paper	43	APS-2022-06-25-OR-2	FI00012	1771914070	OFFICE	in-ST			
3	Ink cartridge	23	APS-2022-06-25-OR-2	FI00013	2152185755	OFFICE	in-ST			
4	Toilet paper	55	APS-2022-06-25-OR-2	OT0001	4433815865	OTHER	in-ST			
5										
6										

Figure 54: Export outputs example

### 8.3.2 Reading QR Codes

Pressing the "Read QR Codes" button in the menu opens Window #3 enabling you to use the webcam of your computer for scanning purposes. The Youtube video on the following [QR5](#) link demonstrates the functionality of this feature. I recommend watching the video before further reading.



QR 5: Video 2 – Reading QR Codes

The biggest button "Start Scanning" dominates Window #3 (Figure 55). Upon clicking this button, another window is created for projecting the video capture of the computer's webcam (Figure 56). The button "Show the last WSS.db entry:" serves exactly that purpose. If clicked, the label under the button will show the record of the last database entry. In Figure 55, the last entry is shown as an empty list [], as the database had been deleted and no scanning followed afterward. The red button allows the deletion of all of the database's content.

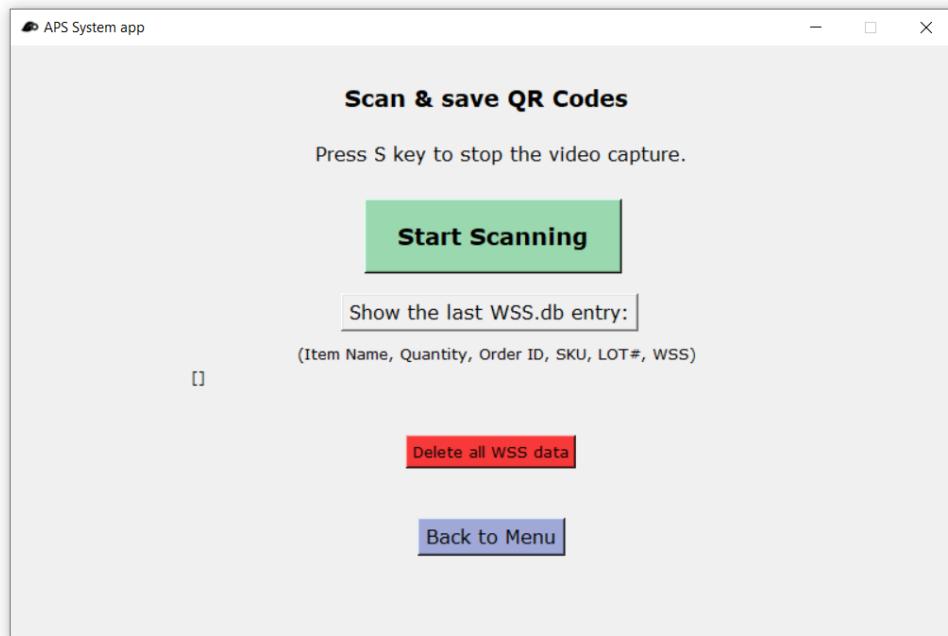


Figure 55: APS App QR reading window

The "Back to Menu" button is kindly taking the user back to the menu.

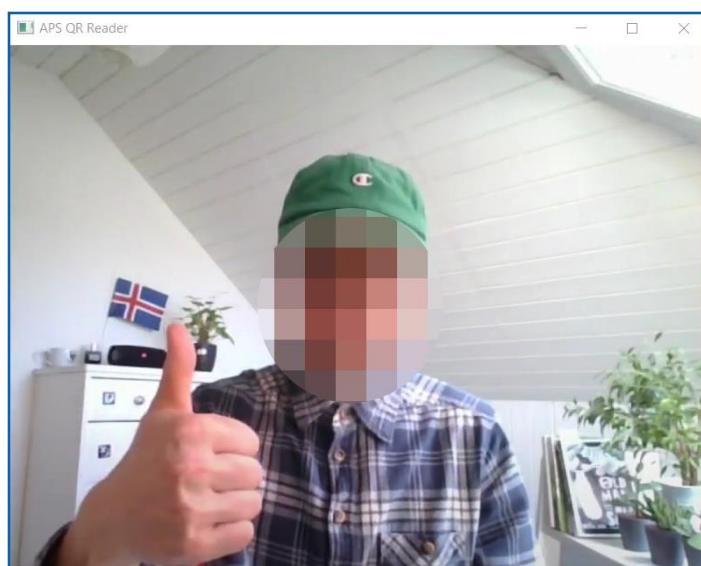


Figure 56: Video Capture window

This functionality works as a tool for stocking sub-process mentioned in Chapter 8.2. Logistics staff will transport the pallet with packages to be stored in particular WSS. Then they should scan the WSS ID label that the item should be stored in. This could potentially be enhanced for each shelf having its dedicated QR identifier, but for now, each WSS ID label is representing the whole rack or storing area. The labels used in the demonstration video are shown in Figure 57. After the WSS label is scanned, the package QR label is scanned, and the package is placed in the storing location. This will immediately update the database. This task should be made for every package entering the company's warehouse.



Figure 57: APS WSS scannable labels

The process described above can then be executed with the usage of the APS System App. When the camera capture window is initialized by the green button in Window #3, the video capture window is opened, and the program starts analyzing the camera captured data. If the QR code is detected in captured screen, the action of a process diagrammed in Figure 58 is triggered. The reading process is running in a loop until the "s" key (representing the "STOP" command) on the keyboard is pressed resulting in the termination of the video capture window.

The program loop in Figure 58 starts reading with the state = 0. If a QR Code is detected, it is checked if it represents one of 5 WSS IDs. If not Message 1 explaining, what type of QR Code is expected will pop up. If the correct WSS ID label is scanned, Message 2 will confirm to the user which WSS is going to be stocked and asks the user to scan the package that should be stored in the previously scanned WSS, and the state is changed to state = 1. This ensures that the WSS ID label can not be scanned as an item. If the package QR Code label is detected it will store its content into the WSS database as the last entry and the Message 3 assures the user that scanning was successful and asks to repeat the process while changing the state back to state = 0 as the WSS ID label is expected to be scanned next.

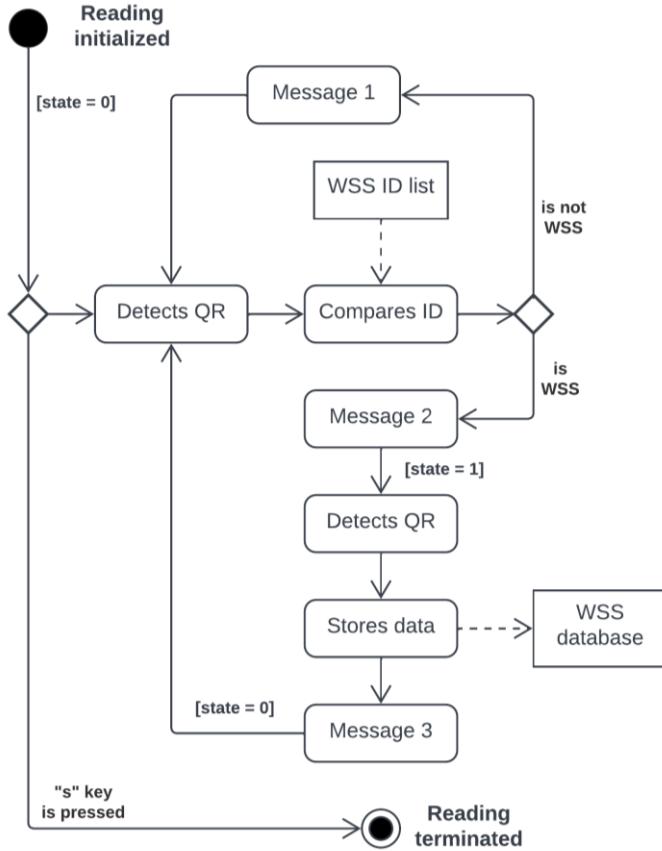


Figure 58: QR Reading code functionality in UML notation

The program works reliably as long as only the 2 types of QR Codes (WSS ID label and Package QR label) are captured by the webcam. In the case of scanning an irrelevant QR code, an error may occur. The infallible solution able of distinguishing not relevant 2D symbols is a subject of future improvement proposals.

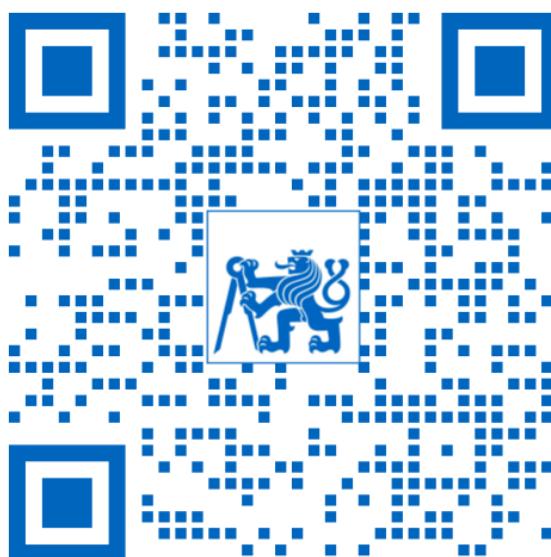
By scanning, the WSS.db database is filled with data from the QR symbols. The example of this database filled with 10 entries is in Figure 59. [92]

WSS (10 rows)					
<code>SELECT * FROM 'WSS' LIMIT 0,30</code>					
Item_Name	Quantity	Order_ID	SKU	LOT	WSS
Controlunit	3	APS-2022-06-25-OR-1	EL00011	1347542816	WSS-A2
nWheelelectricmotor	12	APS-2022-06-25-OR-1	EL00021	1143585376	WSS-A1
Weldingmetal	12	APS-2022-06-25-OR-1	ME00390	1489574888	WSS-WS
Cuttingelectricmotor	9	APS-2022-06-25-OR-1	EL00015	1707439044	WSS-A2
MetalCuttings	42	APS-2022-06-25-OR-1	ME00004	4772275343	WSS-WS
Fasteners	420	APS-2022-06-25-OR-1	ME00499	6322092027	WSS-A1
Oxygenbottle	5	APS-2022-06-25-OR-1	CG00033	1391642385	WSS-CG
Officepaper	43	APS-2022-06-25-OR-2	FI00012	1771914070	WSS-ST
nkcartridge	23	APS-2022-06-25-OR-2	FI00013	2152185755	WSS-ST
Toiletpaper	55	APS-2022-06-25-OR-2	OT0001	4433815865	WSS-ST

Figure 59: WSS Database content

### 8.3.3 Support Functionalities

APS System App Menu has 3 more buttons. The most relevant of the three is the “Import New Data” button. Pressing it opens Window #2 (Figure 60) enabling rewriting of the supplied parts database. This database serves as a data source for the APS Item Ordering System. As already mentioned, this feature has a rather supportive character and is not directly used in the proposed process described in Chapter 8.2. The Youtube video on the following [QR6](#) link demonstrates the functionality of this feature. This video is not necessary for the new proposed solution comprehension.



QR 6: Video 3 - Support Functionalities

Window #2 (Figure 60) includes a text label with a description of the desired format of the data that are about to be imported. The left button opens “Open File Dialog”, where the file directories can be browsed, and desired Excel file selected. The right button will start a program procedure, where old

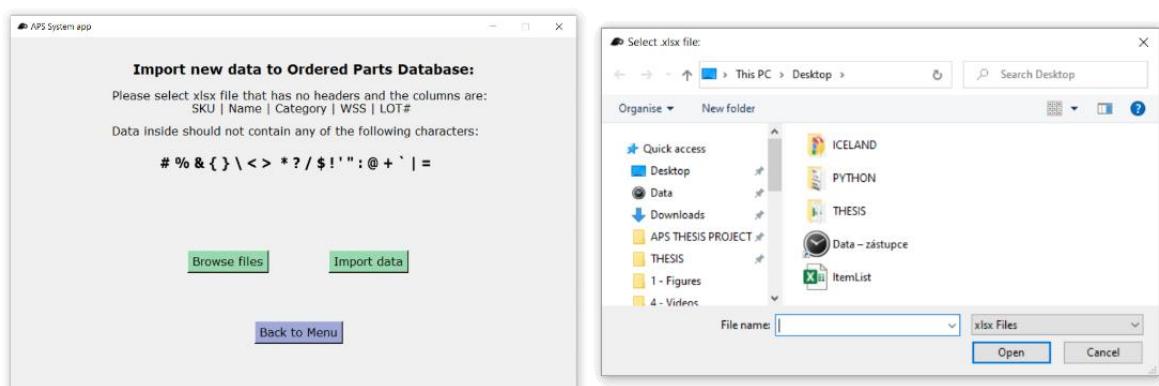


Figure 60: APS App New Data Import windows

data from the database are removed, the uploaded Excel file is converted into CSV format, and the database is filled with converted data. After the upload, the APS System App should be restarted to reflect these changes.

The remaining Menu buttons will lead to the GUI windows but lack any functionality. AR Demo in Window #4 (Figure 61A) was intended to use OpenCV together with ArUco markers to demonstrate a use case for augmented reality. This idea is closer introduced in Chapter 9.2.

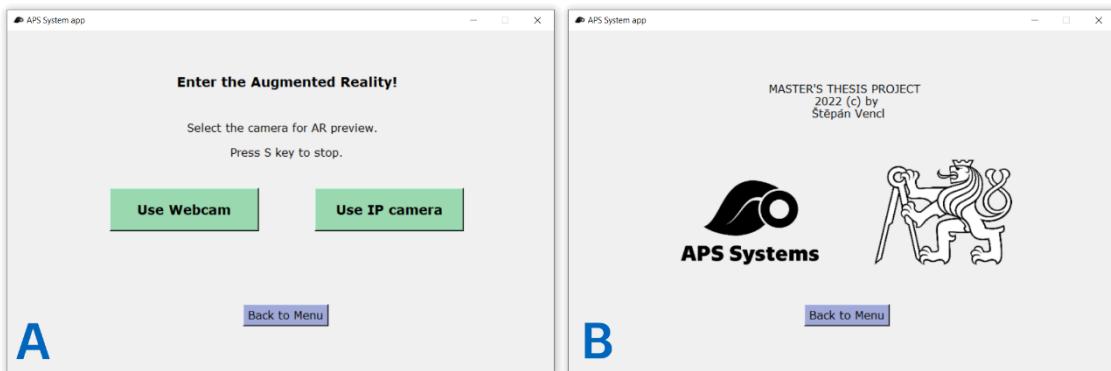


Figure 61: APS App Windows #4 and #5

The last Window #5 (Figure 61B) introduces the author to the user and summarizes the purpose of this application as well as shows the magnificent CTU lion as well as the APS Systems logo.

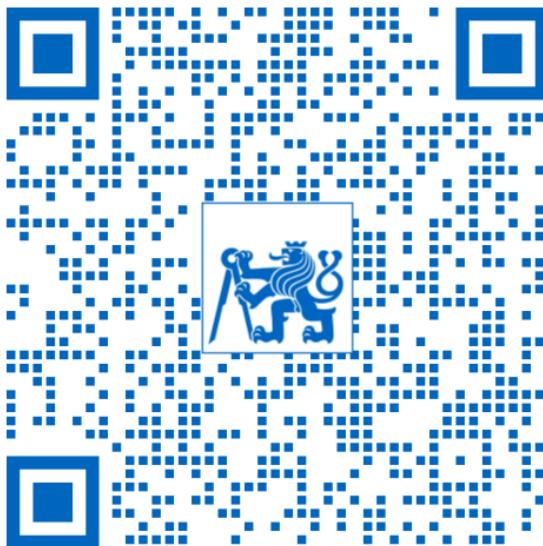
## 8.4 Code overview

The code for this Python application was written by me and only myself. Considering me being a coding beginner, I am very well aware of the imperfections in my code.

Some parts of the code could have been made more neatly or efficiently, sometimes the code is repeated. And I know there is a better way to code these, for example by using functions, or classes and their methods – as in object-oriented programming language should have been done. My way of doing things was goal, not technique motivated. Using simple coding principles at the cost of code redundancy. I simply do not have the experience yet to be able to evaluate what is the best solution for each functional part of the code.

Despite these imperfections, I am more than happy that I successfully created something that works and connects the real world with the virtual,

binary one. I published the project publicly on the GitHub repository accessible on the following [QR7](#) link.



QR 7: APS Systems App project repository

#### 8.4.1 Project structure

When starting a new project in PyCharm IDE, the SW creates a new directory, where all project-related files are stored. Some of such folders and files serve just for Python for different purposes. PyCharm also creates a virtual environment(venv) for the project. This venv stores exact versions of all used extensions by the time project was coded, to prevent any version compatibility issues in the future. If we neglect these directories and special files used by Python or the PyCharm, the structure of my APS System App project is schematically illustrated in Figure 62 below.

Directory **Media** stores all the WSS labels that are pasted on top of the generated QR Code during the order export to create enhanced QR Codes. All logos for the GUI are also stored here.

Directory **Data** is used to store the **ItemListData** database containing information about supplied parts. Also, a CSV file that is an intermediate product of a new data import to the database is stored here.

Directory **Export** contains all the folders with exported orders. Upon every single finished order new folder is created as described in Chapter 8.3.1.

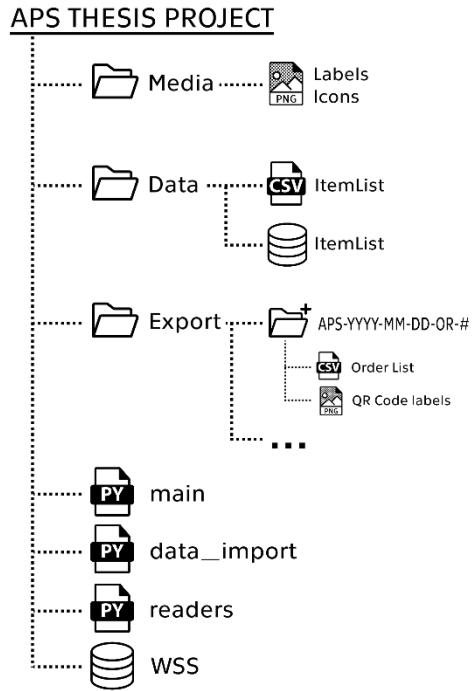


Figure 62: APS App project structure

The main directory contains all three Python files. The code of the program could be in a single file, but this division serves for easier legibility of the code. The **main.py** contains most of the functions and GUI specifications. The **data\_import.py** contains functions related to creating and managing databases. The **readers.py** file contains code working with the OpenCV library which utilizes the webcam of the device. The last file in the directory is the **WSS** database. Here all the information about scanned items is stored.

### 8.4.2 GUI

As already mentioned, the usage of a Graphical User Interface environment is necessary for every SW today. Programs are used throughout the GUI by regular common users. It brings intuitiveness and prevents human errors.

The program for 6 GUI windows is coded within the **main.py** file, using the Tkinter library. An example of this code is included in Attachment 13.

- Window #0 = App Menu window
- Window #1 = New Order window
- Window #2 = Import New Data window
- Window #3 = Read QR Codes window
- Window #4 = AR Demo window
- Window #5 = About window

The theory of the Technology Acceptance Model (TAM) indicates that the Perceived Ease of Use drastically impacts the actual use of new technology. And GUI does unarguably increase this ease of use. [86]

### 8.4.3 Functions

It does not make sense to put all the code here in normal text, nor in the form of screenshots, since the program contains hundreds of lines. The code itself contains comments clarifying the purpose of each bigger segment of the code in relative detail. curious minds can access the code via the browser on GitHub (QR7 link) where the file can be opened, viewed, and browsed.

As an example, Figure 63 shows a selected function from the main.py file, to illustrate the Python syntax and my work:

```
172     # Iterate and gather data for generating QRs, paste labels based on storing spot
173     def iterate_qr():
174         read_csv = pd.read_csv(file_name)
175         for index, values in read_csv.iterrows():
176             global qr_data, qr_orid, qr_name
177             qr_name = values['Item Name']
178             qr_qty = values['Quantity']
179             qr_orid = values['Order ID']
180             qr_sku = values['SKU']
181             qr_lot = values['Supplier lot#']
182             spot = values['Storing spot']
183             qr_data = f'''
184                 Item: {qr_name}
185                 Quantity: {qr_qty}
186                 Order ID: {qr_orid}
187                 SKU: {qr_sku}
188                 Lot #: {qr_lot}
189                 ...
190                 if spot == 'in-A1':
191                     create_qr('SS A1 b')
192                 elif spot == 'in-A2':
193                     create_qr('SS A2 b')
194                 elif spot == 'in-WS':
195                     create_qr('SS WS b')
196                 elif spot == 'in-CG':
197                     create_qr('SS CG b')
198                 elif spot == 'in-ST':
199                     create_qr('SS ST b')
200                 else:
201                     messagebox.showerror(title='Error', message='Iterating went wrong :( ')
```

Figure 63: Code Example

Function `iterate_qr()`: is indented in function `finish_order()`: that is triggered by clicking the button "Finish the order" in GUI Window #1. Inside the `finish_order()`: prior to the displayed example, the outer self-explanatory function `create_folder()`: is called and a folder named after the newly generated Order ID. Then the order preview is checked and if it is empty, the program throws an error with a message popup window. If the order contains at least one item, a CSV file with added header is generated out of these data and is saved to a newly formed folder, and is as well named after the previously generated Order ID. Still indented in `finish_order()`: function, another function called `create_qr(label)`: is defined. This function compiles the QR Code symbol out of fixed parameters, the current content of the global `qr_data` variable and at the end pastes the WSS label based on the passed `label` argument through the parenthesis of the function and saves the QR symbol as a .png file to the created folder. Finally, we are getting to the displayed `iterate_qr()`: function description. The previously generated CSV list is read and using for loop stores values from columns to the local variables for each iteration. After filling the variables, the previously described function `create_qr(label)`: is called where the `label` argument differs determined by the `spot` value (desired WSS) inside the conditional statements. After the displayed code segment in the Figure 63 example, the function itself is called. After execution, the pop-up message will inform the user about the exported files and adds 1 to the index global variable `order_id`. As a last step of the function `finish_order()`: is to return the GUI view to the menu Window #0. The indentation ends.

As can be noticed from the description of a small segment of the code stated in the example, a broader context is always needed. The whole program is interconnected through various variables and functions. Attachment 14 shows more of the code and functions in the PyCharm IDE environment. I consider these examples sufficient for demonstration purposes.

# 9 Further Improvements

This Chapter introduces all the thoughts on the possible upgrades and improvements for the APS System App or for the proposed process solution. Some of them were briefly mentioned in previous Chapters and here are elaborated on further.

## 9.1 App Upgrades

### Detail of locators

The most impactful upgrade of the current solution is in my opinion enhancing the detail of the storage location. Instead of 5 WSS scannable labels dedicated QR Code can be made for each column/row of the storing rack or specific label locator within each WSS. Scanning these can bring so much more detail to the inventory traceability and can eliminate staff time loss during the search for components. Figure 64 shows a visualization of this improvement idea with a unique label for 3 storing spots on each shelf level. But each compartment could also have its own QR for the most detailed location traceability. The question is, what the best ratio of clarity and complexity of such system is.

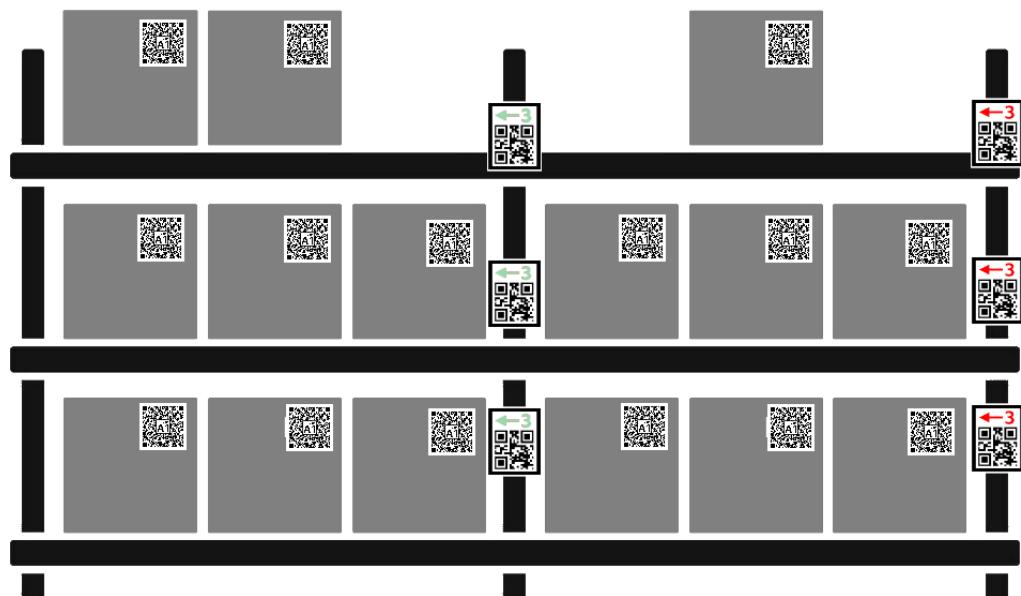


Figure 64: WSS rack with scannable location ID labels

## **IS integration**

The improvement on a higher-system level might be the integration of this system into the company's ERP, Customer Relationship Management, or Supply Chain Management systems. The ultimate goal is to automate administrative tasks, so the human work capacity can be focused on solving problems, that computers or AI still can not solve.

## **Order form detail**

Another improvement could be adding more detail to the ordering system form. The company has different suppliers for different types of goods and components. This supplier info should be added to the company data structure and reflected in the ordering system window.

The ordering system window could also fetch the data from the WSS database for the user to know, how much of the item is currently in inventory.

## **Scanning improvements**

Reading of QR codes functionality should be made more secure as the irrelevant 2D codes can be scanned by mistake and fill the database with nonsensical data. This requires implementing conditions to the code, that are able to identify the data structure of QR Code encoded data to exclude irrelevant scans together with informing the operator of the system.

During the stocking, the items can be assigned to any WSS that is scanned, despite their intended storing location, displayed on the QR label. The system could be improved to question the operator if the item is assigned to the correct WSS. In these exceptions, the operator would realize the mistake or could enter justifying comments that would be written to the database.

Data structure could also be enhanced by adding the date and time of scanning (registering) the data to the WSS. This would further increase traceability and allow investigation of some deviations if occurred.

## **9.2 AR Demo**

An idea to include AR Demo functionality in my App was to demonstrate, how augmented reality can be used in the enterprise process because most people think AR is a domain of the entertainment industry only.

I mentioned one example of utilizing AR in Chapter 3.5.3. Similarly, AR could be utilized in every industrial company, that is trying to upgrade its processes through digitalization.

Training, maintenance, and visualization for customers are the most potent fields of utilizing AR in industrial companies, at least in my opinion. For example, a trainee could preview the assembly procedure through AR App on a smartphone step by step. A maintenance worker or repairman can access the datasheets and manuals about the device that is to be repaired. Salesmen can amaze potential customers with interactive AR product visualizations or business cards. The App could potentially include and interconnect all suitable process enhancements proposed above. [87]

I was not able to realize my demo idea due to the compatibility issues of the OpenCV library in Python, but I will present the vision of how I wanted to demonstrate AR use within my application. I generated, printed, and placed ArUco markers on each drawer of a dresser in my room. This dresser was meant to represent different compartments and part trays on an assembly workstation. These markers could then be detected by a webcam of a laptop or through IP webcam app, that allows connecting your smartphone camera with a running OpenCV program on your computer. This program would then overlay the captured image with shapes, text, or labels, and the ArUco markers would serve as position anchors and reference points. Figure 65 shows A) the original photo, B) the visualization of detecting the markers and assigning text to them and highlighting borders, and C) the visualization of the possible use case on a fictional assembly workstation. [88]

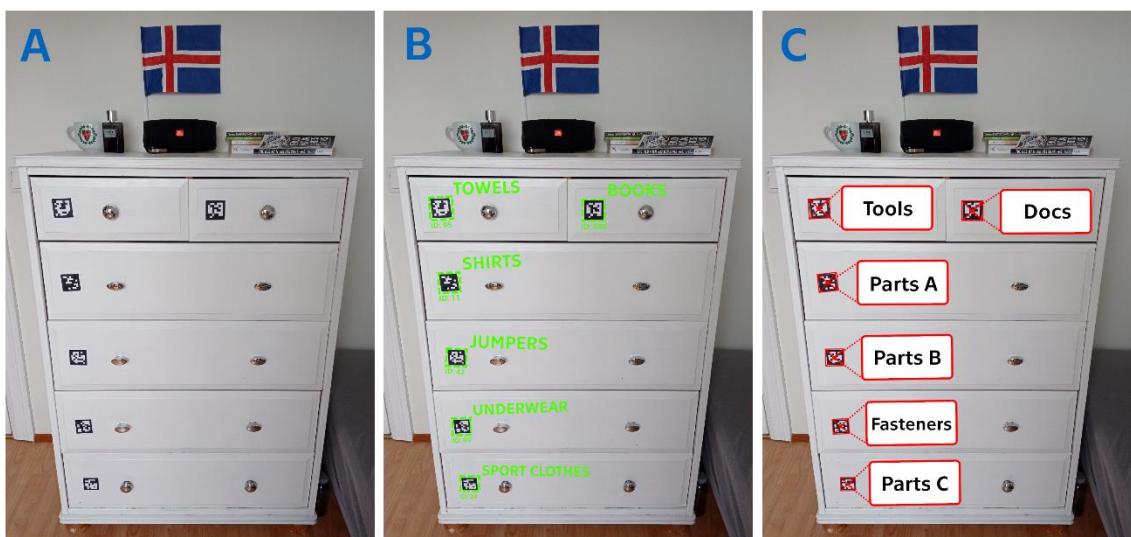


Figure 65: AR Demo visualization variants

### 9.3 UI and UX Designs

User Interface (UI) and User Experience (UX) designs play a major role in the success of any SW solution. UI allows us to control SW in a virtual world, which is then able to control HW in the real world. UI can be voice-enabled, command-line interface, or nowadays the most common - GUI. UX is then based on users' needs and uses the phenomena of human psychology. Some studies suggest that every single detail plays a role in our decision-making. For example, colors can influence our object perception. [89]

TAM (see Chapter 8.4.2) already showed that we need to have a little bit of both. If we build an application with the best functionality but the usability is not clear, the users are going to be less willing to use it. At the same time, if the application has the best UI and UX but it lacks any purpose, the users' behavior will not be any more welcoming. That is why it is a good idea to reflect these principles in any SW or process design. It would be worth considering whether to make some adjustments reflecting these principles to my APS Systems App as well. [86][89][90]

The ultimate vision in my mind employing all topics mentioned in this thesis is a smartphone-compatible application serving as an "in-hand tool" - an extension of the company's IS. It could have accounts connected to each working role or position within the company with different access permissions and responsibilities. It could allow access, view, and share data straight from the source, ease and speed up communication. Assigning and tracking tasks could help with project management and clarify an overview of inter-department projects. Management would have better tools helping them to develop employees' potential and to motivate them. The company's performative objectives could be shared to suppress the silo mentality mentioned in Chapter 1. User authentication could enable passing responsibilities in different stages of processes and eliminate redundant paperwork. It could also support various processes as my thesis demonstrates using built-in HW components. The employees could be concurrently available in the physical and virtual world, and it might lead to bridging the gap between reality and the data "on the paper".

Nowadays companies use various specialized tools to support their processes and for data transition to the digital world. Specialization often

goes hand-in-hand with a higher price tag. But this cost is not always necessary.

The usage of the smartphones during last 10 years became ubiquitous. These devices became the entry ticket to the virtual world even for people in developing countries. Smartphones became the main platform via through users access the internet as the graph in Figure 66 shows. [91]

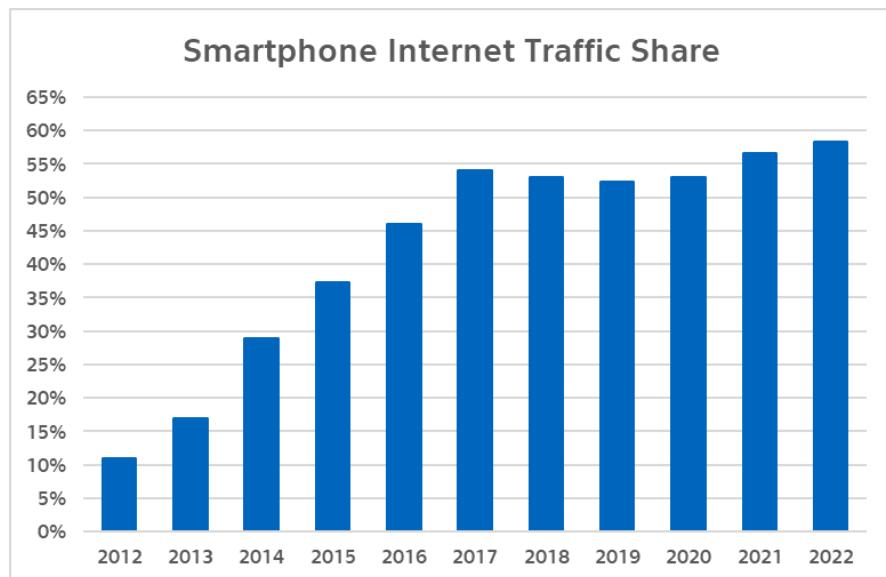


Figure 66: Smartphones internet traffic share. [91]

The success of a smartphone lies in its affordability, portability, connectivity, and computing power comparable with cheaper laptops. It has a camera, and a big enough screen to be easily operable. For all these reasons I tend to think, that incorporating these universal devices into digitalized enterprises and their processes will be a future trend. That is why developing smartphone-compatible SW applications interconnected with an enterprise's Information Systems might be a good idea.

## Conclusion

Processes are a vital part of every enterprise. The aim throughout this thesis has been to introduce possible options helping with process efficiency and prove the possibility and usefulness of utilization of the 2D codes in these processes. Specifically, I was dealing with utilizing QR Codes to improve logistic and inventory management problem in the company APS Systems.

The initial Chapters dealt with the topics of processes, their management, and modeling. Subsequently, I described 1D optical codes as an evolutionary precursor of the 2D codes - representatives of which I described in detail, and finally compared them with other AIDC technologies. I also introduced software tools that later helped me to fulfill the set goals.

The company APS Systems, its background, products, and processes were described. All previously obtained knowledge helped me to target an inefficient process, which was concerning the logistics and stocking of supplied parts and insufficient traceability of this stocked inventory.

As an improvement proposal, I modeled and designed an improved process and created a process map, which illuminates the operation and the flow of the process to all participants and eliminates any unclarities. The desktop application designed for ordering and registering supplied parts written in Python programming language serves as a support tool for the execution of this proposed process.

My application can generate special QR Codes for labeling the packages of the supplied parts ordered through the application. The reading functionality is then employed in the stocking sub-process, resulting in higher traceability and an overall smoother transition of the supplied parts into production. To demonstrate my application, I created 3 videos with a detailed description of the application's functionalities. Although the application is functional, it could be subject to further development. In addition to introducing the features of the application and how to work with it, I also presented the code behind it and introduced possible improvements, specific suggestions for future extensions and other ideas. The coding project and other related documents are available online on the GitHub repositories.

I consider the set goals to be fulfilled.

## A Abbreviations & Acronyms

Abbreviation	Full Definition
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
A1	Assembly 1
A2	Assembly 2
AI	Artificial Intelligence
AIDC	Automatic Identification and Data Capture
AR	Augmented Reality
ARIS	Architecture of Integrated Information Systems
ASCII	American Standard Code for Information Interchange
B&W	Black and White
B2B	Business to Business
B2C	Business to Consumer
BI	Business Intelligence
BP	Business Process
BPD	Business Process Diagram
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPMN	Business Process Model and Notation
BPMS	Business Process Management Systems
BPR	Business Process Reengineering
BR	Barcode Reader
CAPA	Corrective and Preventive Action
CASE	Computer Aided Systems Engineering
CCD	Charge-Coupled Device
CG	Chemicals & Gases
CMOS	Complementary Metal-Oxide Semiconductor
CSV	Comma-Separated Values
CTU / ČVUT	Czech Technical University in Prague
CV	Computer Vision
CW	Codeword
DC	Data Character
DM	Data Matrix
EAN	European Article Number
EBIT	Earnings Before Interest and Taxes
EC	Error Correction
EEC	Enhanced Error Correction
EIS	Enterprise Information System
EP	Enterprise Process
ERP	Enterprise Resource Planning
EU	European Union
FM	Fiducial Marker
FP&A	Financial Planning and Analysis
GTIN	Global Trade Item Number
GUI	Graphical User Interface
HCCB	High Capacity Color Barcode
HQ	Headquarters
HW	Hardware

<b>Abbreviation</b>	<b>Full Definition</b>
IAN	International Article Number
IC	Integrated Circuit
ID	Identification Number
IDE	Integrated Development Environment
IS	Information System
ISBN	International Standard Book Number
ISO	International Organization for Standardization
ITF	Interleaved 2 of 5
KB	Kilobytes
LED	Light-Emitting Diode
LSB	Least Significant Bit
MS	Microsoft
MSB	Most Significant Bit
NA	North America
NFC	Near Field Communication
OCR	Optical Character Recognition
OMG	Object Management Group
OOP	Object-Oriented Programming
PDCA	Plan-Do-Check-Act
PEP	Python Enhancement Proposals
POS	Point of Sale
PS	Paint Shop
QC	Quality Control
QR	Quick Response
RFID	Radio Frequency Identification
RS	Reed-Solomon
RU	Reykjavík University
SCM	Structured Carrier Message
SEC	Standard Error Correction
SKU	Stock Keeping Unit
SPC	Statistical Process Control
SQL	Structured Query Language
SW	Software
TAM	Technology Acceptance Model
ToC	Table of Contents
TQM	Total Quality Management
UI	User Interface
UID	Unique Identification
UML	Unified Modeling Language
UPC	Universal Product Code
URL	Uniform Resource Locator
US / USA	United States of America
UX	User Experience
WS	Welding Shop
WSS	Warehouse Storing Spot

## B Bibliography & References

- [1] Claeys, B. (2021). *The free QR code generator to create custom QR codes with logo*. QRCode Tiger. Retrieved February 2022, from [link](#)
- [2] Icelandic Government. (2016). *Icelandic National Flag*. Retrieved March 2022, from [link](#)
- [3] Kavan, M. (2018) *Organizace a řízení výroby* [personal lecture notes]. CTU in Prague, Faculty of Mechanical Engineering, Department of Enterprise Management and Economics
- [4] Cambridge Dictionary. (2022). *Process definition: Meaning of process in English*. Retrieved March 2022, from [link](#)
- [5] Sharp, A., & McDermott, P. (2009). *Workflow Modeling: tools for process improvement and applications development*. (2nd ed.). Boston: Artech House. ISBN 978-1-59693-192-3
- [6] Kantnerová, L., Stašák, J., & Petrášková, V. (2016). *Procesní řízení a modelování s přihlédnutím k praxi v logistice*. Jihočeská univerzita v Českých Budějovicích. ISBN 978-80-7394-598-5
- [7] Svozilová, A. (2011). *Zlepšování podnikových procesů*. Grada. Expert (Grada). ISBN 978-80-247-3938-0
- [8] Řepa, V. (2012). *Procesně řízená organizace*. Grada. Management v informační společnosti. ISBN 978-80-247-4128-4
- [9] Smith, A. (1827). *An Inquiry into the Nature and Causes of the Wealth of Nations*. University Press. Retrieved March 2022, from [link](#)
- [10] Wyatt, F. J. (2019). *A brief history of process management to the modern-day*. Medium. Retrieved March 2022, from [link](#)
- [11] The W. Edwards Deming Institute. (n.d.). *PDSA Cycle*. Retrieved March 2022, from [link](#)
- [12] Brocke, V. J., & Rosemann, M. (2014). *Handbook on Business Process Management 1*. (2nd ed.). Springer. ISBN 978-3-642-45100-3
- [13] Rouh, M. A. (2018). *Continuous improvement with PDCA*. Medium. Stay curious. Retrieved March 2022, from [link](#)
- [14] Kenton, W. (2020). *Understanding Silo Mentality*. Investopedia. Retrieved March 2022, from [link](#)
- [15] Scott, J. (2016). *Process and Functional Approaches in BPMN*. Good E-Learning Blog. Retrieved March 2022, from [link](#)
- [16] Klimeš, C. (2014). *MODELOVÁNÍ PODNIKOVÝCH PROCESŮ*. Ostravská univerzita v Ostravě. Retrieved February 2022, from [link](#)
- [17] Hammer, M. (1990). *Reengineering Work: Don't Automate, Obliterate*. Harvard Business Review. Retrieved March 2022, from [link](#)
- [18] Řepa, V. (2007). *Podnikové procesy: Procesní řízení a modelování* (2nd ed). Grada. ISBN 978-80-247-2252-8
- [19] Hammer, M. (2007). *The Process Audit*. Harvard Business Review. Retrieved March 2022, from [link](#)
- [20] Šabatová, I. (n.d.). *Podnikové procesy pod kontrolou: Principy a přínosy implementace systémů pro řízení procesů (BPMS)*. Galeos a.s. Retrieved March 2022, from [link](#)
- [21] Rosing, M., Scheel, H., Scheer, A. W., (2014). *The Complete Business Process Handbook*. Elsevier. ISBN 978-0-12-799959-3
- [22] Kotadia, A. (2020). *What is Business Process? – Types, Benefits, and Examples*. Fasproc Blog. Retrieved March 2022, from [link](#)
- [23] International Organization for Standardization (1985). *Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts, and system resources charts* (ISO 5807:1985)

- [24] Thibodeau, P. (2018). *Computer-aided software engineering (CASE)*. TechTarget. Retrieved March 2022, from [link](#)
- [25] ARIS BPM Community. (2021). *BPM Tutorials with ARIS*. Retrieved March 2022, from [link](#)
- [26] Unified Modeling Language. (2005). *What is UML?* Retrieved April 2022, from [link](#)
- [27] Ave Coders. (2021). *All UML Diagrams in 10 minutes* [Video]. Retrieved April 2022, from [link](#)
- [28] Object Management Group. (2009). *UML Superstructure Specification, v2.2*. Retrieved April 2022, from [link](#)
- [29] Brtková, P., & Vencl, Š. (2020). *Projektování informačních systémů*. [unpublished semestral assignment] ČVUT.
- [30] Object Management Group (2011). *Business Process Model And Notation Specification Version 2.0*. Retrieved April 2022, from [link](#)
- [31] BPM Offensive Berlin (2011). *BPMN 2.0 Poster*. Retrieved April 2022, from [link](#)
- [32] Phaniteja, J., & Tom, P.D. (2013). *EVOLUTION OF BARCODE*. ISSN-2320-7884 (Online)
- [33] Kato, H., Tan, K. T., & Chai, D. (2010). *Barcodes for Mobile Devices*. Cambridge University Press. ISBN 978-0-511-71305-7
- [34] Harford Control. (2018). *A Guide to 2D Codes*. Retrieved April 2022, from [link](#)
- [35] Tan, J. S. (2008). *QR Code*. Synthesis Journal, 59–78. Retrieved April 2022, from [link](#)
- [36] Šoustek, P., & Matoušek, R. (2012). *Moderní čárové kódy*. AUTOMA, 5, 26–29.
- [37] ASCII Code. (n.d.). *The extended ASCII table*. Injosoft. Retrieved April 2022, from [link](#)
- [38] Cognex. (2021). *INTRODUCTION TO INDUSTRIAL BARCODE READING*. Retrieved April 2022, from [link](#)
- [39] TEL Nanotec Museum. (n.d.). *What Is a CMOS Image Sensor?* Retrieved April 2022, from [link](#)
- [40] Apps on Google Play. (n.d.). *Scandit Barcode Scanner Demo*. Retrieved April 2022, from [link](#)
- [41] Dynamsoft Blog. (2020). *The Comprehensive Guide to 1D and 2D Barcodes*. Retrieved April 2022, from [link](#)
- [42] GS1. (2022, February 21). *About / GS1*. Retrieved April 2022, from [link](#)
- [43] Barcode Factory. (n.d.). *Free Code 39 Barcode Generator - Generate Your Own Barcode*. Retrieved April 2022, from [link](#)
- [44] GS1. (2022). *Global Trade Item Number (GTIN)*. Retrieved April 2022, from [link](#)
- [45] World Health Organization (2022). *Iceland: WHO (COVID-19) Data*. Retrieved May 2022, from [link](#)
- [46] AIM USA. (n.d.). *PDF417 Specification for Barcode Symbology*. Retrieved May 2022, from [link](#)
- [47] Skoglund, I. (2020). *Reed-Solomon Codes: Error Correcting Codes*. Linnaeus University. [link](#)
- [48] Silver Bay Software LLC. (2009). *Maxicode Encoder Programmer's Manual*. Retrieved May 2022, from [link](#)
- [49] Longacre, A., & Hussey, R. (1995). *Two-dimensional data encoding structure and symbology for use with optical readers* (US5591956A). US Patent and Tm. Office. [link](#)
- [50] Hand Held Products Inc. (2009). *Aztec Code Specification*. Retrieved May 2022, from [link](#)
- [51] Queree, C. (2021). *Barcode Transport Ticketing - An Introduction*. The Ticket Keeper. Retrieved May 2022, from [link](#)
- [52] Microsoft. (2016). *High-Capacity Color Barcodes (HCCB)*. Microsoft Research. Retrieved May 2022, from [link](#)
- [53] ColorCode (2022). *ColorCode Design*. ColorZip.com. Retrieved May 2022, from [link](#)

- [54] GS1. (2018). *GS1 DataMatrix Guideline*. Retrieved May 2022, from [link](#)
- [55] Dumont, T., Lippert, T., Wokaun, A., & Leyvraz, P. (2004). *Laser writing of 2D data matrices in glass*. ScienceDirect. Retrieved May 2022, from [link](#)
- [56] International Organization for Standardization (2015). *Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification* (ISO/IEC 18004:2015(E))
- [57] Avola, D., Cinque, L., Foresti, G. L., Mercuri, C., & Pannone, D. (2016). *A Practical Framework for the Development of Augmented Reality Applications by using ArUco Markers*. International Conference on Pattern Recognition Applications and Methods
- [58] Fiala, M. (2004). *ARTag Revision 1. A Fiducial Marker System Using Digital Techniques*. National Research Council of Canada. NRC 47419/ERB-1117
- [59] Ekdahl, C. (2018). *Aruco Marker Detection*. Wolfram Language Reviews. Retrieved May 2022, from [link](#)
- [60] Gao, J., Kulkarni, V., Ranavat, H., Chang, L., & Mei H., (2009) *A 2D Barcode-Based Mobile Payment System*. IEEE Conference Publication. DOI: 10.1109/MUE.2009.62
- [61] Querini, M., & Italiano, G. F. (2013). Facial recognition with 2D color barcodes. *International Journal of Computer Science and Applications*. 10. 78-97.
- [62] Berners-Lee, T., Masinter, L., & McCahill, M. (1994). *Uniform Resource Locators (URL)*. RFC 1738. Retrieved May 2022, from [link](#)
- [63] Claeys, B. (2022). *Difference between static and dynamic QR code*. QRtiger. Retrieved May 2022, from [link](#)
- [64] GS1 (2022). *GS1 Element Strings Guideline*. Retrieved May 2022, from [link](#)
- [65] Sorensen, E. (2021). *QR code payments - what is it and how does it work?* Retrieved May 2022, from [link](#)
- [66] De Pace, F., Manuri, F., Sanna, A. (2018). *Augmented Reality in Industry 4.0*. iMedPub Journals. DOI:10.21767/2349-3917.100017
- [67] Schneider Electric. (2021). *Rozšířená realita v praxi – EcoStruxure Augmented Operator Advisor*. Vše o průmyslu. [link](#)
- [68] IBM Cloud Education. (2022). *What Is Optical Character Recognition (OCR)?* Retrieved May 2022, from [link](#)
- [69] FIS Global. (2019). *The Evolution of Credit Cards*. Worldpay from FIS – Insights. Retrieved May 2022, from [link](#)
- [70] IBM. (n.d.). *Magnetic Stripe Technology*. Retrieved May 2022, from [link](#)
- [71] Hyman, V. (2021). Swiping left on magnetic stripes. Mastercard. Retrieved May 2022, from [link](#)
- [72] JCB Europe. (2013). *About IC Chip Cards / JCB Card*. Retrieved May 2022, from [link](#)
- [73] Maestre, S. (2021). *What is an IC and Why is There a Massive Global Chip Shortage?* CircuitBread. Retrieved May 2022, from [link](#)
- [74] Atlas RFID Store. (n.d.). *The beginner's guide to RFID systems*. Retrieved May 2022, from [link](#)
- [75] U.S. Food and Drug Administration. (2018). *Radio Frequency Identification (RFID)*. Retrieved May 2022, from [link](#)
- [76] RFID4U. (2020). *RFID Tag construction - what goes into an RFID tag or label*. Retrieved May 2022, from [link](#)
- [77] Wankhede, C. (2022). *What is NFC and how does it work?* Android Authority. Retrieved May 2022, from [link](#)
- [78] MattKC. (2020). *Can you fit a whole game into a QR code? [Video]*. Retrieved May 2022, from [link](#)

- [79] Python Software Foundation. (2022). *Python*. Python.Org. Retrieved June 2022, from [link](#)
- [80] Wikipedia contributors. (2022). *Python (programming language)*. Wikipedia. Retrieved June 2022, from [link](#)
- [81] JetBrains s.r.o. (2021). *PyCharm: the Python IDE for Professional Developers*. JetBrains. Retrieved June 2022, from [link](#)
- [82] GitHub. (n.d.). *GitHub: Where the world builds software*. Retrieved June 2022, from [link](#)
- [83] Lucid Software Inc. (n.d.). *What is Business Process Modeling Notation? Lucidchart*. Retrieved June 2022, from [link](#)
- [84] Makhmatov, A., Brtková, P., & Vencl, Š. (2020). *Projekt 3* (lead by Ing. Miroslav Žilka, Ph.D.). [semestral assignment] ČVUT.
- [85] Shopify Encyclopedia. (n.d.). *Stock Keeping Unit (SKU)*. Shopify. Retrieved June 2022, from [link](#)
- [86] Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). *User Acceptance of Computer Technology: A Comparison of Two Theoretical Models*. Management Science, 35(8), 982–1003. <https://doi.org/10.1287/mnsc.35.8.982>
- [87] Makarov, A. (2022). *AR in Retail, Marketing, and Sales in 2022*. MobiDev. Retrieved June 2022, from [link](#).
- [88] Kalachev, O. (n.d.). *Online ArUco markers generator*. Chev.Me. Retrieved May 2022, from [link](#)
- [89] Babich, N. (2021). *UI vs. UX Design: The Similarities & Differences*. Adobe XD. Retrieved June 2022, from [link](#)
- [90] Bramo, I., Fasca, L., Magnus, K., & Reis, A. (2012). *The Contribution of Color to Object Recognition*. Advances in Object Recognition Systems. <https://doi.org/10.5772/34821>
- [91] Gaubys, J. (2022). *What Percentage of Internet Traffic Is Mobile?* Oberlo. Retrieved June 2022, from [link](#)
- [92] Novák, J. (2020). *SQLite Viewer*. GitHub.io. Retrieved June 2022, from [link](#)

## C Lists

### List of Figures

Figure 1: Iceland Example brackets. Based on [2].....	11
Figure 2: PDCA iterative cycle. [13] .....	13
Figure 3: Silo Mentality and vertical org. structure. Based on [15]. .....	14
Figure 4: Sharp & McDermott's Process enablers [5].....	16
Figure 5: Process Types. Based on [21] .....	18
Figure 6: Flowchart symbols in MS Word .....	19
Figure 7: ARIS House. Based on [25].....	20
Figure 8: Notation for UML Activity diagram [28].....	22
Figure 9: BPMN 2.0 Event types [31] .....	23
Figure 10: BPMN 2.0 Activities [31].....	24
Figure 11: BPMN 2.0 Task types [31].....	24
Figure 12: BPMN 2.0 Activity markers [31].....	24
Figure 13: BPMN 2.0 Gateway types [30] .....	25
Figure 14: BPMN 2.0 Data elements [30].....	25
Figure 15: BPMN 2.0 Sequence Flow types [31] .....	25
Figure 16: BPMN 2.0 Connecting Objects [30] .....	26
Figure 17: BPMN 2.0 Pool with 2 Swimlanes [12].....	26
Figure 18: Correct way to capture workflow within the pool [5].....	26
Figure 19: BPMN 2.0 Artifacts [31].....	27
Figure 20: 2D vs 1D code [34] .....	28
Figure 21: Barcode evolution [35].....	30
Figure 22: Laser & Image-based readers [38].....	33
Figure 23: Classification of 1D code symbologies [33] .....	34
Figure 24: Code 39 symbol. Based on [33].....	35
Figure 25: EAN-13 symbol. Based on [33] .....	36
Figure 26: Code 128 symbol. Based on [33] .....	37
Figure 27: Barcode I received for PCR testing .....	38
Figure 28: PDF 417 symbol. Based on [33][36] .....	40
Figure 29: MaxiCode symbol. Based on [33][48].....	41
Figure 30: Aztec Code symbol. Based on [43][50].....	42
Figure 31: Old bus ticket verification.....	43
Figure 32: New bus ticket verification.....	44
Figure 33: HCCB types and comparison [52].....	45

Figure 34: Data Matrix symbol. Based on [33][54].....	46
Figure 35: QR Code symbol. Based on [33][56] .....	47
Figure 36: QR Code demasking. Based on [56] .....	48
Figure 37: AR marker symbols [57][58].....	50
Figure 38: Static and Dynamic symbols. Based on [63].....	53
Figure 39: RFID system. Based on [74] .....	55
Figure 40: SW tools logos [79][81][82][83].....	58
Figure 41: APS Systems Organizational Structure [84] .....	64
Figure 42: APS Meower [84] .....	65
Figure 43: APS System EU distribution network [84] .....	66
Figure 44: Manufacturing process of APS Meower [84].....	67
Figure 45: APS facility floor plan .....	68
Figure 46: WSS labels .....	69
Figure 47: Inventory flow schematics in floorplan detail.....	70
Figure 48: Order ID structure .....	73
Figure 49: Encoding order data into QR label .....	74
Figure 50: APS supply chain process in BPMN 2.0 notation .....	76
Figure 51: APS System App Menu window.....	77
Figure 52: Supplied Parts Database content.....	78
Figure 53: APS App Item Ordering System window.....	79
Figure 54: Export outputs example .....	80
Figure 55: APS App QR reading window.....	81
Figure 56: Video Capture window .....	81
Figure 57: APS WSS scannable labels.....	82
Figure 58: QR Reading code functionality in UML notation .....	83
Figure 59: WSS Database content .....	83
Figure 60: APS App New Data Import windows.....	84
Figure 61: APS App Windows #4 and #5 .....	85
Figure 62: APS App project structure.....	87
Figure 63: Code Example .....	88
Figure 64: WSS rack with scannable location ID labels .....	90
Figure 65: AR Demo visualization variants .....	92
Figure 66: Smartphones internet traffic share. [91] .....	94

## List of Tables

Table 1: QR Code EC levels [56] .....	48
Table 2: 2D codes capacity comparison [33][36][50] .....	51
Table 3: AIDC technologies comparison. Based on [33][43][...] .....	57
Table 4: APS System logo & information [84] .....	63

## List of QR links

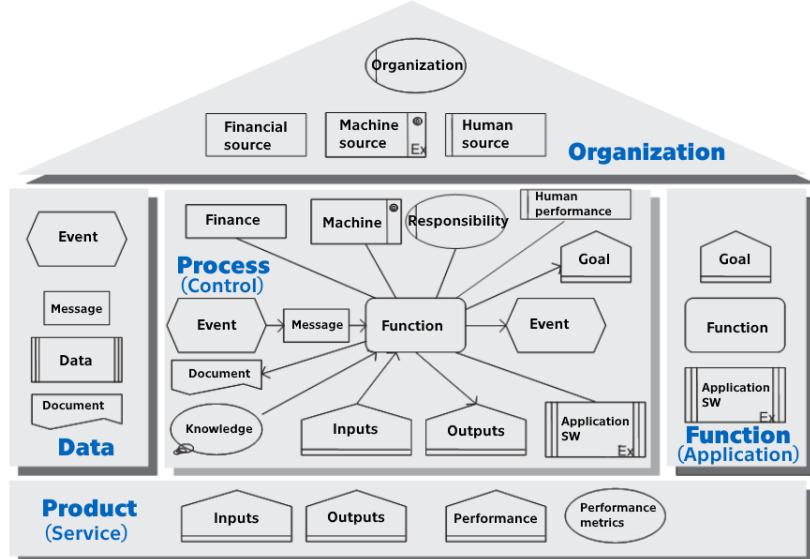
QR 1: External link example [1] .....	11
QR 2: Scandit Demo – BR Mobile Application [40].....	33
QR 3: APS Systems documentation [84] .....	62
QR 4: Video 1 – Creating New Order.....	78
QR 5: Video 2 – Reading QR Codes.....	80
QR 6: Video 3 - Support Functionalities.....	84
QR 7: APS Systems App project repository.....	86

## List of Attachments

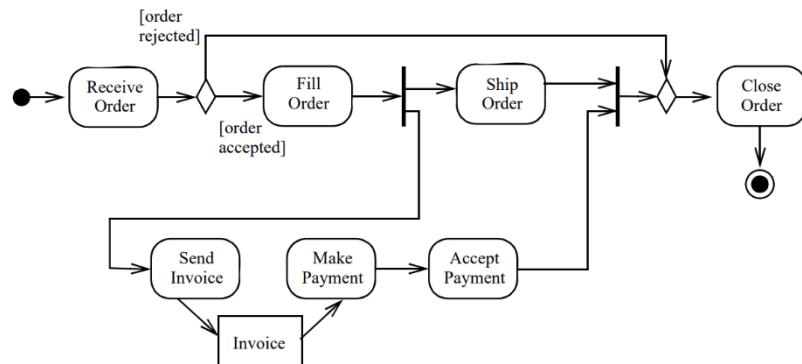
Attachment 1: ARIS House example [18].....	105
Attachment 2: UML activity diagram example [28] .....	105
Attachment 3: UML class diagram example. Based on [28] .....	105
Attachment 4: UML sequence diagram example [29] .....	106
Attachment 5: BPMN 2.0 event types [31].....	106
Attachment 6: PDF 417 Codeword detail [46] .....	106
Attachment 7: Aztec symbology detail [49][50] .....	106
Attachment 8: Data Matrix laser engraved micro symbol [55] .....	106
Attachment 9: QR Code masking patterns [56] .....	106
Attachment 10: QR Code decoding sequence in UML notation [56] .....	106
Attachment 11: RFID gate in Landsbókasafn Íslands .....	106
Attachment 12: APS System supplied parts list [84].....	106
Attachment 13: Example of GUI section of the code.....	106
Attachment 14: Example of function section of the code.....	106

## D Attachments

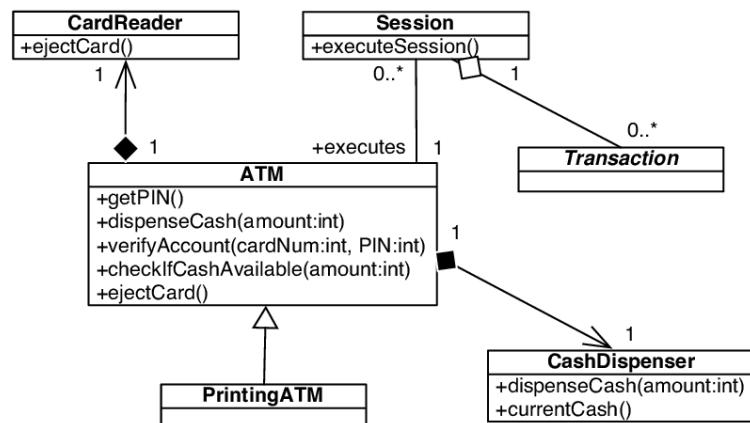
Attachment 1: ARIS House example [18]



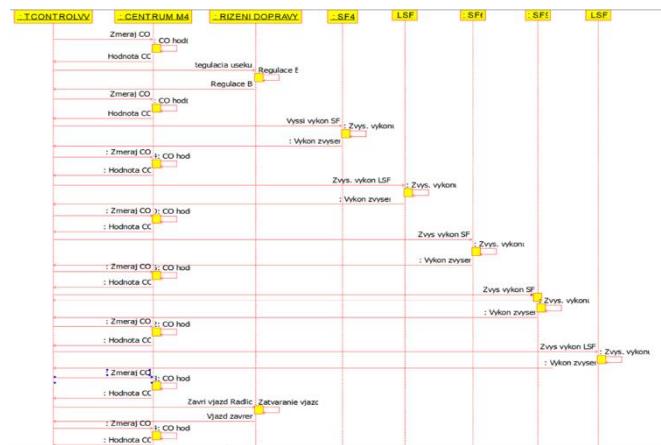
Attachment 2: UML activity diagram example [28]



Attachment 3: UML class diagram example. Based on [28]



Attachment 4: UML sequence diagram example [29]

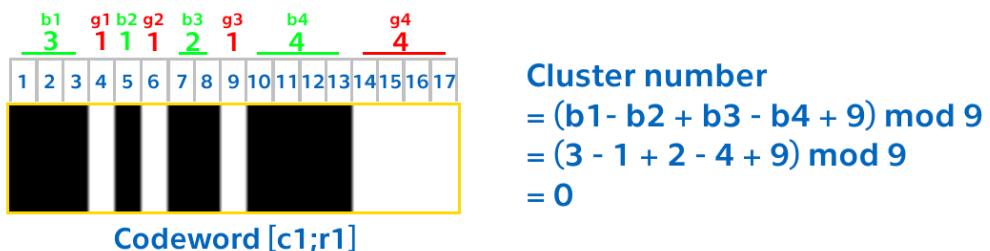


Attachment 5: BPMN 2.0 event types [31]

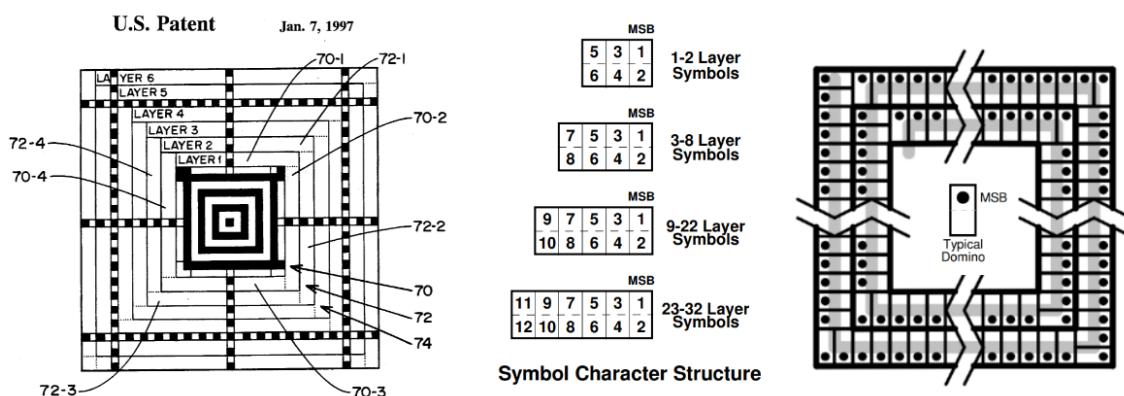
## Events

	Start	Intermediate	End				
Standard	Event Sub-Process Interrupting	Event Sub-Process Non-Interrupting	Catching	Boundary Interrupting	Boundary Non-Interrupting	Throwing	Standard
<b>None:</b> Untyped events, indicate start point, state changes or final states.	(empty circle)					(empty circle)	(empty circle)
<b>Message:</b> Receiving and sending messages.	(envelope icon)	(envelope icon)	(envelope icon)	(envelope icon)	(envelope icon)	(envelope icon)	(envelope icon)
<b>Timer:</b> Cyclic timer events, points in time, time spans or timeouts.	(clock icon)	(clock icon)	(clock icon)	(clock icon)	(clock icon)	(clock icon)	(clock icon)
<b>Escalation:</b> Escalating to an higher level of responsibility.	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)
<b>Conditional:</b> Reacting to changed business conditions or integrating business rules.	(rectangle icon)	(rectangle icon)	(rectangle icon)	(rectangle icon)	(rectangle icon)		
<b>Link:</b> Off-page connectors. Two corresponding link events equal a sequence flow.			(arrow icon)			(arrow icon)	
<b>Error:</b> Catching or throwing named errors.	(wavy line icon)		(wavy line icon)			(wavy line icon)	
<b>Cancel:</b> Reacting to cancelled transactions or triggering cancellation.				(cross icon)			(cross icon)
<b>Compensation:</b> Handling or triggering compensation.	(left arrow icon)		(left arrow icon)	(left arrow icon)		(left arrow icon)	(left arrow icon)
<b>Signal:</b> Signalling across different processes. A signal thrown can be caught multiple times.	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)	(triangle icon)
<b>Multiple:</b> Catching one out of a set of events. Throwing all events defined	(pentagon icon)	(pentagon icon)	(pentagon icon)	(pentagon icon)	(pentagon icon)	(pentagon icon)	(pentagon icon)
<b>Parallel Multiple:</b> Catching all out of a set of parallel events.	(plus icon)	(plus icon)	(plus icon)	(plus icon)	(plus icon)	(plus icon)	
<b>Terminate:</b> Triggering the immediate termination of a process.							(solid circle)

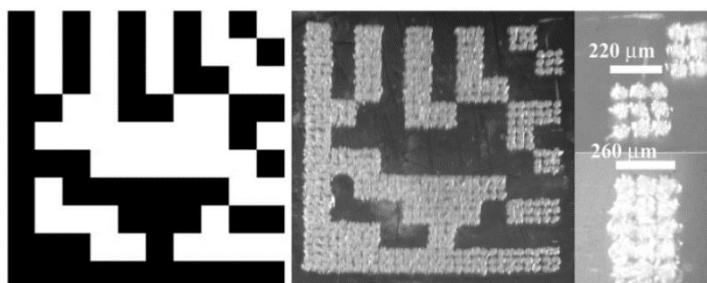
### Attachment 6: PDF 417 Codeword detail [46]



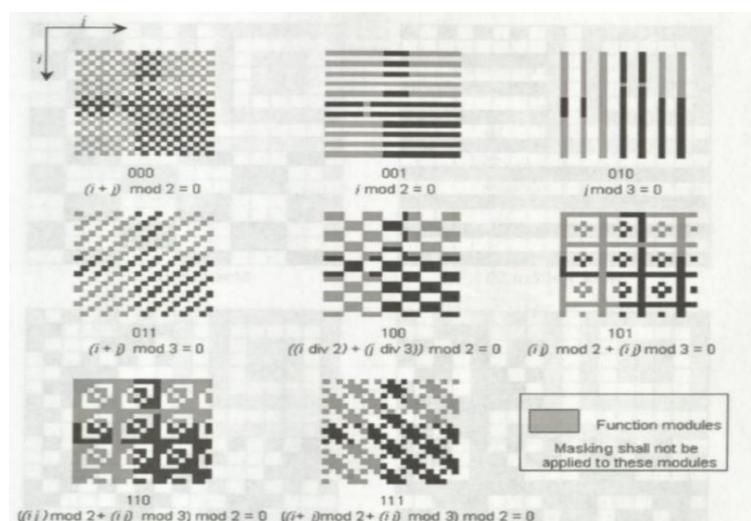
### Attachment 7: Aztec symbology detail [49][50]



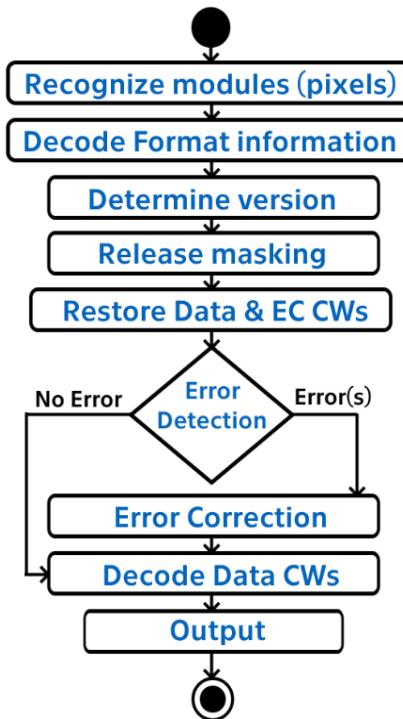
### Attachment 8: Data Matrix laser engraved micro symbol [55]



### Attachment 9: QR Code masking patterns [56]



Attachment 10: QR Code decoding sequence in UML notation [56]



Attachment 11: RFID gate in Landsbókasafn Íslands



## Attachment 12: APS System supplied parts list [84]

Original part name	parts per product	SKU	English part name	Category	Storing spot	Supplier lot #
Elektromotor - inwheel	4	EL00021	InWheel electric motor	ELECTRO	in-A1	1143585376
Elektromotor - stepper	4	EL00026	Stepper electric motor	ELECTRO	in-A2	8375540703
Elektromotor - rotor	3	EL00015	Cutting electric motor	ELECTRO	in-A2	1707439044
Řídící jednotka	1	EL00011	Control unit	ELECTRO	in-A2	1347542816
Baterie	10	EL00032	Battery packs	ELECTRO	in-A2	9764407469
Výstražná světla	4	EL00124	Warning lights	ELECTRO	in-A2	3346675090
Alarm	1	EL00123	Alarm buzzer	ELECTRO	in-A2	6327267685
Wifi reciever and transmitter	1	EL00070	Wifi reciever & transmitter	ELECTRO	in-A2	5083061159
GPS module	1	EL00079	GPS module	ELECTRO	in-A2	3519322445
Koncové sensory	18	EL00280	Limit switches	ELECTRO	in-A2	5172742735
RADAR sensors	4	EL00057	RADAR sensors	ELECTRO	in-A2	2448437221
Optic Camera sensors	2	EL00056	Optic Camera sensors	ELECTRO	in-A2	8505694875
Elektrohydraulika	1	EL00099	Electrohydraulics	ELECTRO	in-A1	4961846640
Charging Dock	1	EL00042	Charging Dock	ELECTRO	in-A1	5982507342
Kabeláž	20	EL00345	Cables & connectors	ELECTRO	in-A2	8023147586
Ostatní elektronika	1	EL00399	Other electronics	ELECTRO	in-A2	8562538591
Výpalky	15	ME00004	Metal Cuttings	METAL	in-WS	4772275343
Výkovky	3	ME00019	Metal Forgings	METAL	in-WS	5658033012
Ohýbané plechové díly	12	ME00022	Bent sheet metal parts	METAL	in-WS	1395244075
Obrobky	10	ME00025	Steel Workpieces	METAL	in-WS	5652600009
Nože	3	ME00002	Cutting blades	METAL	in-A1	3412174473
Písty	4	ME00121	Hydraulic Pistons	METAL	in-A1	1850588435
Pájecí cín	NA	ME00370	Solder tin	METAL	in-A2	3361593270
Svařovací kov	NA	ME00390	Welding metal	METAL	in-WS	1489574888
Spojovací materiál	1	ME00499	Fasteners	METAL	in-A1	6322092027
Koš	1	PL00004	Collecting bag	PLASTIC	in-A1	9800279605
Bumpers	4	PL00013	Plastic Bumpers	PLASTIC	in-A1	5075928491
Odolné funkční díly	5	PL00016	Durable functional parts	PLASTIC	in-A1	1253134722
Pohledové díly	8	PL00021	Visual parts	PLASTIC	in-A1	5908386410
Kola	4	PL00001	Wheel tires	PLASTIC	in-A1	1571459728
Ostatní plastové díly	1	PL00099	Other plastic parts	PLASTIC	in-A1	4836485625
Základová barva	NA	CG00011	Primer color	CH&G	in-CG	1960230445
Povrchová barva	NA	CG00012	Surface color	CH&G	in-CG	1011370700
Kyslik	NA	CG00033	Oxygen bottle	CH&G	in-CG	1391642385
CO2	NA	CG00035	CO2 bottle	CH&G	in-CG	9838015509
Ar-He	NA	CG00036	Ar-He bottle	CH&G	in-CG	7996258323
Čistící chemikálie	NA	CG00088	Cleaning chemicals	CH&G	in-CG	2091736885
Ostatní chemické přípravky	NA	CG00089	Other chemicals	CH&G	in-CG	1603048011
Kancelářský papír	NA	FI00012	Office paper	OFFICE	in-ST	1771914070
Inkoustová kazeta	NA	FI00013	Ink cartridge	OFFICE	in-ST	2152185755
Psací potřeby	NA	FI00014	Writing accessories	OFFICE	in-ST	2532457440
Pásky	NA	FI00015	Tapes	OFFICE	in-ST	2912729125
Kancelářské nástroje	NA	FI00016	Office tools	OFFICE	in-ST	3293000810
Složky a pořadače	NA	FI00017	Folders and binders	OFFICE	in-ST	3673272495
Ostatní kancelářské potřeby	NA	FI00018	Other office supplies	OFFICE	in-ST	4053544180
Toaletní papír	NA	OT0001	Toilet paper	OTHER	in-ST	4433815865
Sanitární potřeby	NA	OT0002	Sanitary needs	OTHER	in-ST	4814087550
Hnojivo pro rostliny	NA	OT0003	Fertilizer	OTHER	in-ST	5194359235
Ostatní	NA	OT0004	Other	OTHER	in-ST	5574630920

## Attachment 13: Example of GUI section of the code

```

303 # ===== Frame 0 code = Menu Screen
304
305 # Label
306 f0_spacer = Label(frame0, text="")
307 f0_spacer.pack()
308 f0_title = Label(frame0, text="APS System App Menu:", font=APSFont)
309 f0_title.pack(pady=10)
310
311 # MENU Buttons
312 f0_btn1 = Button(frame0, text="New Order", command=lambda: new_order(),
313                 width=16, height=2, bd=6, bg=APScolor, font=APSBold, relief=GROOVE)
314 f0_btn1.pack(pady=5)
315 f0_btn2 = Button(frame0, text="Import New Data", command=lambda: show_frame(frame2),
316                 width=16, height=2, bd=6, bg=APScolor, font=APSBold, relief=GROOVE)
317 f0_btn2.pack(pady=5)
318 f0_btn3 = Button(frame0, text="Read QR Codes", command=lambda: show_frame(frame3),
319                 width=16, height=2, bd=6, bg=APScolor, font=APSBold, relief=GROOVE)
320 f0_btn3.pack(pady=5)
321 f0_btn4 = Button(frame0, text="AR Demo", command=lambda: show_frame(frame4),
322                 width=16, height=2, bd=6, bg=APScolor, font=APSBold, relief=GROOVE)
323 f0_btn4.pack(pady=5)
324 f0_btn5 = Button(frame0, text="About", command=lambda: show_frame(frame5),
325                 width=16, height=2, bd=6, bg=APScolor, font=APSBold, relief=GROOVE)
326 f0_btn5.pack(pady=5)
327
328 # ===== Frame 1 code = Ordering system
329
330 # Title Label
331 f1_space = Label(frame1, height=1)
332 f1_space.grid(row=0, column=2, columnspan=2)
333 f1_title = Label(frame1, text="APS Item Ordering System:", font=APSBold)
334 f1_title.grid(row=1, column=2, columnspan=2, pady=15)
335 # Label 1.1
336 f1_label1 = Label(frame1, text="Search items to be ordered by name:", font=APSLight)
337 f1_label1.grid(row=2, column=0, columnspan=2, padx=15, pady=5)
338
339 # Combobox
340 sql_name = ()
341 fetch_name()
342 combo = ttk.Combobox(frame1, values=sql_name, width=30, height=15, font=APSLight)
343 combo.grid(row=3, column=0, columnspan=2, padx=15, pady=10)
344
345 # Label 1.2
346 f1_label2 = Label(frame1, text="Enter quantity to be ordered:", font=APSLight)
347 f1_label2.grid(row=2, column=2, columnspan=2, padx=15, pady=5)
348
349 # Quantity entry field
350 quantity = Entry(frame1, width=15, font=APSLight)
351 quantity.grid(row=3, column=2, columnspan=2, padx=15, pady=10)
352
353 # Button 1.1
354 f1_btn1 = Button(frame1, text="Add to the order list", command=lambda: add_order(), font=APSLight, bg=APScolor)
355 f1_btn1.grid(row=3, column=4, pady=10)
356
357 # TreeView of the order
358 tree = ttk.Treeview(frame1)
359 tree[ 'columns' ] = ( 'Item Name', 'Quantity', 'Order ID', 'SKU', 'Supplier lot#', 'Category', 'Storing spot' )
360 tree.column( '#0', width=35, minwidth=30, anchor= CENTER )
361 tree.column( 'Item Name', width=150, minwidth=100, anchor= W )
362 tree.column( 'Quantity', width=70, minwidth=50, anchor= CENTER )
363 tree.column( 'Order ID', width=150, minwidth=50, anchor= CENTER )
364 tree.column( 'SKU', width=80, minwidth=50, anchor= CENTER )
365 tree.column( 'Supplier lot#', width=120, minwidth=100, anchor= CENTER )
366 tree.column( 'Category', width=80, minwidth=50, anchor= CENTER )
367 tree.column( 'Storing spot', width=80, minwidth=50, anchor= CENTER )
368
369 tree.heading( '#0', text='#', anchor= CENTER )
370 tree.heading( 'Item Name', text='Item Name', anchor= CENTER )
371 tree.heading( 'Quantity', text='Quantity', anchor= CENTER )
372 tree.heading( 'Order ID', text='Order ID', anchor= CENTER )
373 tree.heading( 'SKU', text='SKU', anchor= CENTER )
374 tree.heading( 'Supplier lot#', text='Supplier lot#', anchor= CENTER )
375 tree.heading( 'Category', text='Category', anchor= CENTER )

```

## Attachment 14: Example of function section of the code

```

52     # Enabling to have multiple "windows" in GUI
53     def show_frame(frame):
54         frame.tkraise()
55
56
57     for frame in (frame0, frame1, frame2, frame3, frame4, frame5):
58         frame.grid(row=0, column=0, sticky="NESW")
59
60     show_frame(frame0)
61
62
63     # Obtaining data from the db to display in combobox (dropdown list)
64     def fetch_name():
65         connection = sqlite3.connect("Data/ItemListData.db")
66         cursor = connection.cursor()
67         cursor.execute('select Part_Name from PARTS')
68         global sql_name
69         sql_name = cursor.fetchall()
70         connection.close()
71
72
73     # Onclick: Views Frame 1, creates new order id and clears previous order data from the treeview
74     def new_order():
75         global order_num
76         order_num = "APS-" + str(today) + "-OR-" + str(order_id)
77         show_frame(frame1)
78         print(order_num)
79         for item in tree.get_children():
80             tree.delete(item)
81
82
83     # Obtaining remaining data about selected item in combobox
84     def fetch_remaining(x):
85         connection = sqlite3.connect("Data/ItemListData.db")
86         cursor = connection.cursor()
87         name = (x, )
88         qry = "select SKU, Category, Storing, LOT from PARTS where Part_Name=?"
89         cursor.execute(qry, name)
90         global data
91         data = cursor.fetchall()
92         connection.close()
93
94
95     # Onclick: Adds selected item to the Treeview as a preview for the order
96     def add_order():
97         global i
98         qty = quantity.get()
99         selected = combo.get()
100
101        if qty == "" or selected == "":
102            messagebox.showerror(title='Error', message='Please select the item and quantity')
103        elif qty.isdigit() == False:
104            messagebox.showerror(title='Error', message='Enter integer as a quantity!')
105        elif qty != "" and selected != "":
106            cleared = selected.strip('{}')
107            fetch_remaining(cleared)
108            tree.insert(parents'', index='end', iid=str(i), text=str(i),
109                        values=(cleared, qty, order_num, data[0][0], data[0][3], data[0][1], data[0][2]))
110            quantity.delete(0, END)
111            combo.delete(0, END)
112            i += 1
113        else:
114            messagebox.showerror(title='Error', message='Unexpected error x0: ')
115
116
117     # Onclick: Deletes selected items in Treeview from the order preview
118     def remove_selected():
119         r = tree.selection()
120         for record in r:
121             tree.delete(record)
122
123
124
125     # Creates folder in Export directory for particular order. Overwrites if one already exists
126     def create_folder():
127         global order_num
128         dirpath = os.path.join('./Export/', order_num)

```