ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pikous** Jméno: **Tomáš** Osobní číslo: **420340**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Softwarové řešení pro zvýšení efektivity práce v rozsáhlém průmyslovém prostředí

Název diplomové práce anglicky:

Software solution for optimizing work efficiency in large-scale industrial environment

Pokyny pro vypracování:

1.Prostudujte state of the art možnosti zvýšení efektivity a potenciální redukce lidských chyb při obsluze průmyslových balících linek.
2.Navrhněte architekturu a systém, který zjednoduší a zefektivní identifikaci jednotlivých balených materiálů, provede pracovníka balícím procesem a umožní zaznamenání práce pracovníků.
3.Podle předchozího návrhu implementujte systém a otestujte v praxi.
4.Analyzujte získaná data a porovnejte výkonnost se současným procesem. Proveďte testování s uživateli za podmínek reflektujících pracoviště balící linky.

Seznam doporučené literatury:

1. LASI, H. et al. Industry 4.0. Business & Information Systems Engineering. 2014
2. J. Wan et al., ""Software-Defined Industrial Internet of Things in the Context of Industry 4.0,"" in IEEE Sensors Journal, vol. 16, no. 20, pp. 7373-7380, Oct.15, 2016.
3. doi: 10.1109/JSEN.2016.2565621
4. FIELDING, R. T. Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine, 2000.
5. HONIG, W. Operant behavior: areas of research and application. Century psychology series. Prentice-Hall, 1966
6. NIELSEN, J. Usability Inspection Methods. New York, NY, USA: John Wiley & Sons, Inc., 1994. Heuristic Evaluation

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Martin Klíma, Ph.D., Katedra počítačové grafiky a interakce

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **16.02.2018** Termín odevzdání diplomové práce: ____________

Platnost zadání diplomové práce: **30.09.2019**

[redacted] [redacted] [redacted]

Ing. Martin Klíma, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Master's Thesis

Software solution for optimizing work efficiency in a large-scale industrial environment

Bc. Tomáš Pikous

Supervisor: Ing. Martin Klíma, Ph.D.

Study Programme: Open informatics, Master's degree

Field of Study: Software Engineering

May 23, 2018

Aknowledgements

I would like to thank my advisor, Ing. Martin Klíma Ph.D., for guidance and being helpful, and I would also like to thank my girlfriend and my family for help and support whenever it was necessary.

Declaration

I declare that I elaborated this thesis on my own and that I mentioned all the information sources and literature that have been used in accordance with the Guideline for adhering to ethical principles in the course of elaborating an academic final thesis.

In Prague on May 23, 2018 ....

Abstract

The goal of this work is to design a solution for the reduction of the human error in an internal process of a company offering packaging services because approximately 12% of all packages have some of their contents accidentally swapped. Optical character recognition was selected for item type identification, based on research of currently available solutions on the market and requirements of the packaging company. The intuitive user interface for human operators of the packaging lines with low experience with consumer electronics was designed. The solution is an extension to the current equipment used for packaging, and it is capable of operation under a constrained environment of the packaging line. Logs from packaging process are collected and prepared for analysis using the selected analytic tool. Simple scalability across multiple factories was taken into account during solution design as well as fault tolerance. The solution was successfully tested in a production environment.

Abstrakt

Cílem práce je návrh řešení pro redukci lidských chyb v interním procesu společnosti zajištující balící služby, jelikož u přibližně 12% balení dochází k záměně části obsahu. Metoda optického rozpoznávání znaků byla vybrána pro identifikaci typu balené položky. Výběr proběhl jak na základě analýzy řešení dostupných na trhu, tak na základě požadavků společnosti zajištující balící služby. Bylo navrženo intuitivní uživatelské rozhraní pro obsluhu balících linek, disponující nízkými zkušenostmi s obsluhou elektronických zařízení. Řešení je rozšířením stávajícího vybavení pro kompletaci balení a je schopno pracovat v omezujících podmínkách balící linky. Záznamy z průběhu balícího procesu jsou ukládany a připraveny k analýze za pomoci vybraného analytického nástroje. Jednoduchá škálovatelnost napříč více továrnami, stejně tak jako odolnost proti chybám byly brány v potaz během návrhové fáze. Řešení bylo úspěšně otestování v produkčním prostředí.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter is dedicated to the motivation of the thesis and description of the current state of the problem.

1.1 Motivation

During the past few years, the Internet of Things [13] grows its popularity. It is estimated that there will be up to 40 billion IoT devices connected by 2020 [12]. Those devices are connected and interoperating within existing Internet infrastructure to create smart homes or even cities.

Gartner's hype cycle for emerging technologies (figure 1.1) describes the expectations over time for various technologies. There are huge expectations at the beginning of the life cycle of any invention, followed by rapid decline caused by unsuccessful implementations. If there is a visible progress in the quality, the invention receives investments and reaches plateau phase when the mainstream adopts it.

When hype cycles from past few years are compared, it can be seen that IoT slowly reaches the disillusionment phase caused by fragmentation and non-existence of global standards for communication and interoperability of smart devices. It's estimated that IoT platforms will reach production (plateau) phase in 2 to 5 years with the potential to change the way how the society and industry works.

There are huge expectations of deployment of smart devices in the industrial environment, which is supposed to lead to fourth industrial revolution called Industry 4.0 [25][23]. The cooperation of smart devices in the Industry 4.0 should minimize factory downtime and reduce maintenance expenses, which would lead to productivity increase.

In cooperation with Machine learning principles and Big data analysis, many processes could be optimized or automatized. It will help us to predict failures, optimize manufacturing processes, improve accuracy and therefore result in an economic benefit. Optimization of inefficient processes would reduce wasting resources and therefore reduce environmental impact.

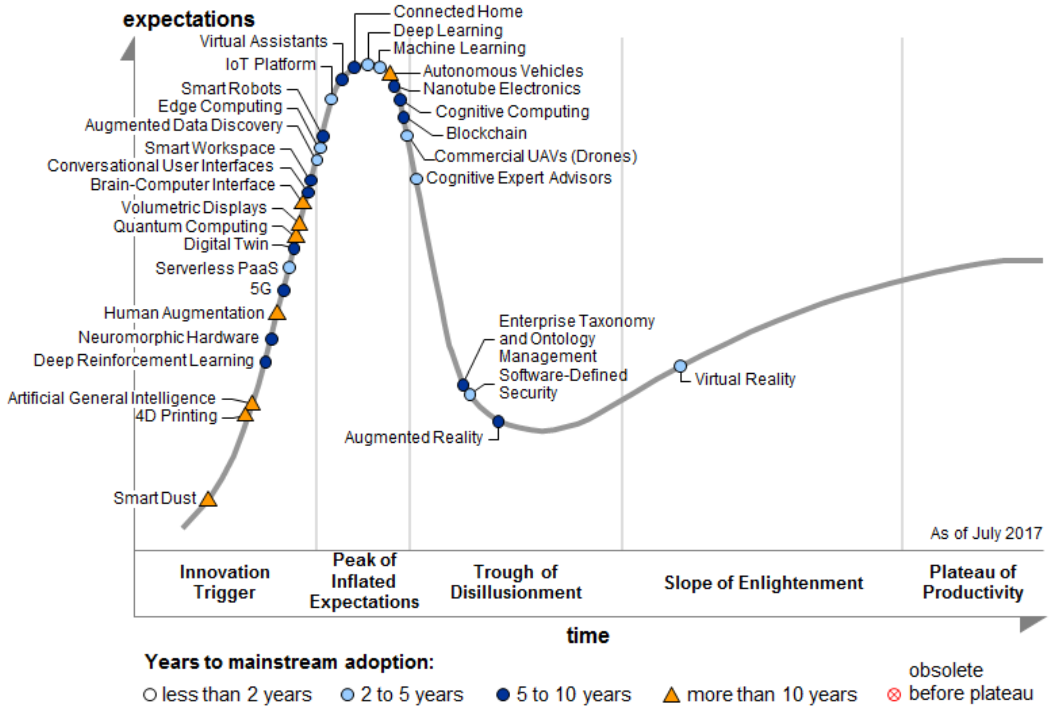

Figure 1.1: Gartner's Hype Cycle for Emerging Technologies, 2017

1.2 Task assignment

This thesis deals with the problem which is a part of a commercial project for a company offering packaging services. They are packaging a large number of various items into separate packages for their customers using old packaging lines without any control or verification mechanisms, and therefore items have to be verified manually by the human operator. The verification process is very archaic, and severe complication occurs when the operator makes a mistake during item type verification. Wrongly packed packages result in financial penalties for the company. The error can be fixed when detected before sending to the customer. The correction of accidental item type substitution takes around six work shifts because approximately 10000 packages need to be unpacked and each item sorted out back to stacks of separate item types, then the packaging process can start over again. This causes a slowdown in the packaging process and therefore financial losses.

1.2.1 Current state

Items are identified by code located at the item itself. This one line code can be up to 80 characters long and contains both alphanumerical and special characters. The contents of a specific package are always described by *packaging instructions*. This document contains the item codes together with their position in the package itself. Items are packed together using machines called *packaging lines*. *packaging line* (figure 1.3) is a large device used to pack large numbers of various items together. This device has up to 16 stackers for each item type, but not all of them must be used in specific packaging instructions. When it is operating, it takes one piece of each item type maintaining their order and packs them together (e.g., using foil wrap, into a box, etc.). This company has tenths of packaging lines, and each of them has one human operator who except for putting the packaging line into operation is responsible for replenishment of stackers.

The operator begins his shift with the initial check of all item types which are delivered to the line by forklift on a pallet. When all delivered item types are corresponding to the packaging instructions, he replenishes the stackers. Each stacker contains items of the same type. Then the packaging line is put into operation, and the packaging process begins. When some of the stackers get empty, the operator stops the machine and replenishes stacker with items of the correct type. When the whole pallet of items is packed, the operator contacts the distribution center via phone and the new pallet with items of requested type is delivered to his packaging line.

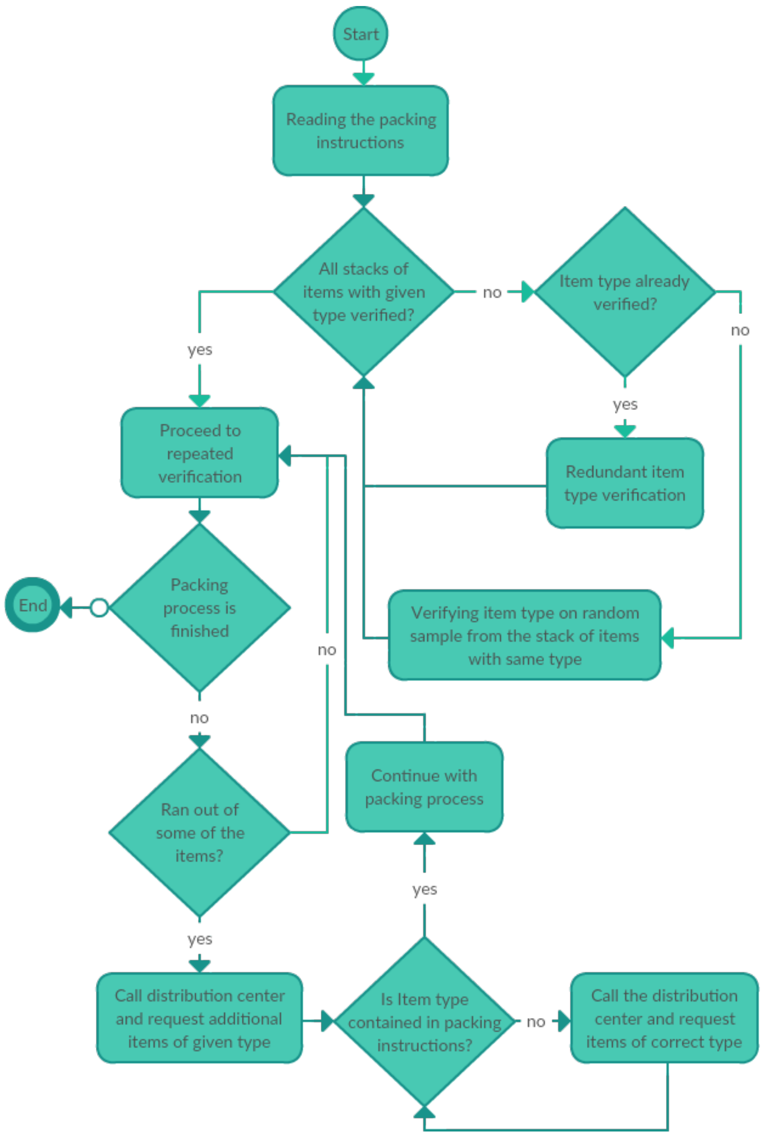

Figure 1.2: Current packaging process

The verification process has many states and multiple loops requiring repetitive work, which may cause that operator will fall into a routine and therefore decreases the amount of attention dedicated to the item type verification and then the error is more likely to happen.

The current solution to keep track of item type verifications is to record all verifications of item types during stacker replenishments on paper. This only helps with finding the author of the mistake but does not prevent mistakes. Currently, the only way ho to prevent errors is to force operators to perform item code verification more carefully via financial penalties.

This is the only control mechanism at the moment, and it is entirely based on negative motivation and repression. According to *operant conditioning* process [19] the balanced mixture of positive and negative motivation is the key to success.

According to the packaging company, the current error rate is approximately 12% of all packed items. There are multiple causes of poor quality item type verification leading to accidental item type substitutions. First is the carelessness of the operator, which means he does not dedicate enough effort to perform verification of sufficient quality. The second cause is the human factor, which means that quality of item verification is reduced due to the characteristics of the human body. For example when the operator is tired (especially during night shifts) or because of overlooking during side-by-side comparison of long item codes (human capability of processing a significant amount of characters).

Another area that could be improved is the communication between an operator and the distribution center, which is currently ensured via standard wire telephone. This process is slow and requires human interaction on both sides. Handling multiple connections is difficult and requires additional human resources.

Each of the mentioned weaknesses causes financial loss and slows down the packaging process. The motivation of the packaging company to improve current state is apparent because there is a possibility to increase revenue by minimizing delays and error occurrence while making the working environment more comfortable for the operators.

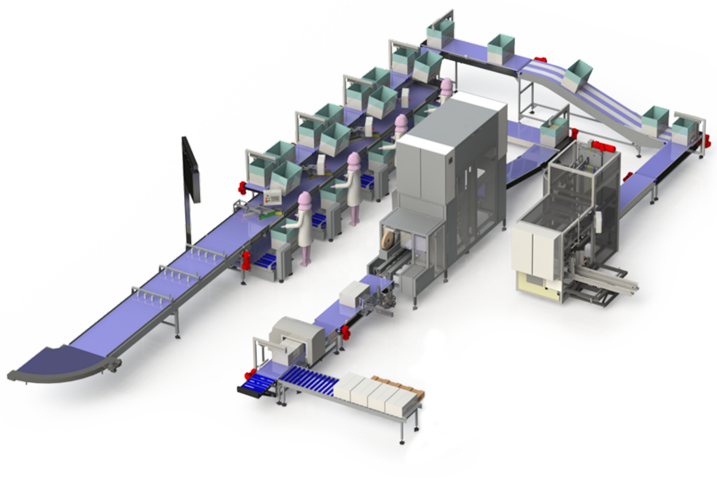

Figure 1.3: Example of packaging line

Chapter 2

Analysis

This chapter is dedicated to the description and analysis of the requirements and constraints for this project.

2.1 System requirements and constrains

This section describes the requirements and constraints given by the packaging company.

2.1.1 Item type verification

The packaging company would like operators to use some electronic device to help them in their tasks, to lower current error rate, which is approximately 12%. The packaging company expects that reading and verifying codes electronically should provide stable results independent of time of day (night shift problem mentioned before) and overall better quality of code verification as machines won't get tired or forget as people do. The goal of the solution is to simplify critical parts in the verification process and transfer the responsibility from operators to machines while minimizing the effort spent from human operators.

2.1.2 Item identifier

The system is required to be able to identify item type and verify whether the item type belongs to package and if the item order is correct. It may use current item type identifier (fig. 2.1 which is the code located on the item itself. Item code has following parameters.

- It is located at the item itself.
- It consists of alphanumeric (both upper and lower case) and special characters (,/: etc.).
- The code is up to 80 characters long.
- The code is always on one line.

Some item types do not have an item type identifier, so the system should be able to allow manual verification of item type.

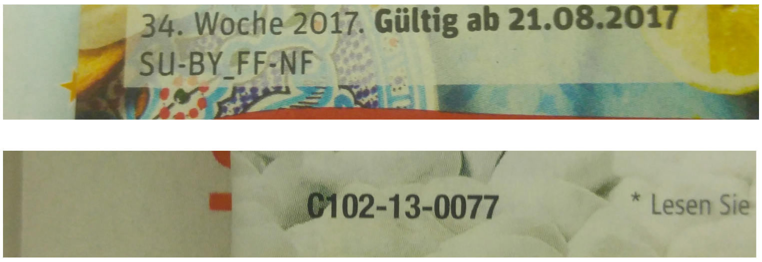

Figure 2.1: Examples of item identifiers used in production

2.1.3 Packaging instructions

The content of a specific package is always described by a document called *packaging instructions*, which defines the *task*. This document contains the item type codes together with their position in the package itself.

To allow automatized verification of item codes, the system should be able to take the output from existing information system as input with minimal modifications of the output format. Currently, the human operator receives a paper document with item types and stacker order specification, generated by the current information system, at the beginning of his work shift.

The system is required to be able to process packaging instructions with up to 30 item codes together with their stacker order.

2.1.4 Analytics and reports

Currently, the logs during the work shift are created manually by the human operator using pen and paper form. When the analysis is required, some employee needs to read through all hand-written logs and process them manually (e.g., enter into information system).

Analysis and performance reports are now done rarely because they require a huge amount of human work. Currently, the logs are processed mainly when something gets wrong, and some item is accidentally replaced with another one. Current state prevents effective work with logs because they are often missing and if not, their processing and analyses are very slow.

The system is required to automatically collect scanning process logs from multiple packaging lines at the same time. It should also provide a way how to access and process collected data.

As the electronic item code verification is the preventive measure, the analysis of the human operator logs from work shift is also desired. The system should be able to support creating reports concerning performance, etc. of the operator. This request also tries to deal with the human factor, in this case, insufficient will to create complete and readable logs.

The system is required to:

- Collect scanning process logs from multiple packaging lines at the same time.

- Use database storage to store and back up the collected logs.
- Support processing the collected data stored in database storage to perform analyses.

2.1.5 User interface

Current employees working as the operators of packaging lines are less educated (elementary school is not an exception) and have low to zero experience with latest electronic devices.

The system should minimize the cognitive load of the human operator while providing an intuitive user interface with the steep learning curve.

2.1.6 Communication and connectivity

Communication between the human operator at packaging line and the distribution center could be improved as well because currently, multiple human operators share one wired telephone. Telephone calls are slowing down the whole process, and the possibility of error occurrence is also higher because the operator requests new palette of items verbally over the phone. This is not the primary goal, but further development in this area in the future should be taken in care in solution design.

There is no local area network currently available in the hall with packaging lines, which may change in the future. Therefore it is required that system is able to work in both cases. When local area network is available, it would also make the communication with the distribution center possible. The requested system should have minimal requirements in this respect.

2.1.7 Other requirements

Scalability
There is currently one hall with packaging lines planned to be equipped with this system, but **expansion to multiple halls or factories must be supported** by the system.

Security
The security measures have to be considered as well, because of possible leakage and abuse of sensitive data and therefore the system should provide a secure way how to store and transfer data between various parts of the system.

Stability
The system is required to withstand loads without losing its functionality to verify codes and collect logs. It is required that system can handle simultaneous logging of packaging process of 16 packaging lines, together with item code verification.

Financial aspect
Current packaging lines are expensive devices fulfilling their primary role with sufficient quality and speed. Therefore the replacement of those devices with newer ones is not considered because of the financial aspect. Requested system should be an addition to current setup, extend it and add requested features with minimal financial costs.

Working conditions
Factory with packaging lines is an industrial environment, which means **difficult working conditions must be taken into account**. Possible risks are mentioned below.

- dust and sand
- extreme humidity
- temperature
- contamination by fluids
- vibrations
- acoustic noise
- shock

2.2 Analysis of system requirements and constrains

This section is dedicated to the analysis of system requirements, constrains and their possible solutions, by utilizing latest technologies.

2.2.1 Item type identification

Key to success in lowering the error rate is according to the packaging company the electronic reading and verification of item identifiers. This method should provide more stable results independent of time of day and overall better quality of code verification. When high-quality identification of item type is needed, the item identifiers should be provided in a machine-readable form. Any information retrievable by any form of energy can be machine-readable. There are several machine-readable medium types allowing to transfer information about item type such as: **Acoustics**, **Electrical**, **Magnetic**, **Optical**.

The according to the nature of items and the item identifiers following medium types were considered:

- Electrical
- Optical

2.2.1.1 Electrical media

Several technologies are using the electrical medium such as memory chips or radio transmission. Technologies using this type of medium, which seems to be suitable for the primary task, are the radio-frequency identification tags (RFID). RFID technology offers standardized way how to store and read data which can be used for item type identification. Tags can be read using wireless reading device their disadvantages are the price because every item has to be equipped with this identifier and the need of configuring the tags attached to the items which bring an additional administrative load.

2.2.1.2 Optical media

Next medium which was considered is the optical medium, containing technologies such as barcodes, QR codes, but also the visual appearance of the item itself. Usage of this medium to provide item type identifier allows easier adaptation of the system to the current state because the additional requirements for customers or the packaging company are lower than in a case with the electrical medium.

Machine readable codes
Utilization of machine-readable codes such as barcodes or QR codes would allow reading the item identifier with almost zero error rate. On the other hand, it would require customers to place this type of machine-readable code at the item itself and provide information about the item types to packaging company, which brings an additional administrative load to packaging company and their customers.

Recognition of visual appearance
Next possibility how to identify item type is to use the visual appearance[16] of the item itself as its identifier. The item would be scanned with a camera to capture its visual appearance and compared with the dataset of appearances of possible item types. There is a huge disadvantage of this method because of the need of creating the dataset of appearances of possible item types, which requires a skilled employee to capture sample images of item type from various angles etc. The administrative load is significantly increased as well as in the previous case.

Recognition of current identifier
According to the requirements and constraints, there is already a unique item identifier in the form of text code located at the item itself which can be used to identify the item type. This would bring almost zero additional requirements and complications to packaging company and their customers.

Current item identifier can be entered into the system using optical character recognition technology which has higher error rate than reading the identifier from machine-readable codes such as barcode or QR code.

The OCR technology provides higher error rate of reading the identifier because the OCR is only a way how to convert human readable format of identifier to a machine-readable format. Despite the fact of the worse accuracy of reading the identifier there is a major advantage in maximal adaptation to the current process.

2.2.1.3 Alternative methods

There are also methods of entering the item identifier into the system using a larger amount of human work such as:

- typing the current item code into machine
- dictating the current item code into machine

The largest disadvantage of those methods is that they are still relying on the human operator to perform the transmission of the item code identifier into the system and therefore they do not eliminate the human factor issue mentioned before. Possible technologies that can be used for voice recognition when dictating the current item identifier are offering good quality of voice recognition. On the other hand, this method still requires error-less reading and also spelling by the operator. The situation is similar to typing the code to machine because it has to be again read and then typed into the machine. In this case, the error probability of mishearing by the voice recognition engine is reduced, but it brings a higher cognitive load to the human operator which is not desired.

2.2.1.4 Conclusion

Previously described methods of inputting the item identifier into the system were divided into two groups after discussion with the packaging company, whether they met their requirements or not.

Rejected methods of reading the item identifier
Solutions relying on a human factor such as manual typing or voice input of item type identifier into the system were found unacceptable. According to the requirements and constraints, the human operators are people with lower education and limited cognitive ability. Mentioned solutions would not lower the possibility of human error, but also bring an additional cognitive load to the human operator.

Solutions significantly **increase administrative load** and therefore were also rejected because of lowering the profitability of the whole project. Identification of item type using the visual appearance is a feasible solution but requires the creation training set of multiple images for each item type which is time-consuming, especially when the item types, packed by the company, are changing rapidly.

Identification by visual appearance would allow identification of item type without any changes in identifiers, which is desired by the customers, on the other hand, there is a significant increase in the amount of work needed to be done by the packaging company even though the error rate is still unacceptable.

Solutions including the addition of machine-readable identifiers to the items are lowering the possibility of identifier read error to almost zero. RFID identifiers are too expensive for this use-case in comparison with machine-readable codes like the barcode or QR code. Unfortunately, they require additional processes to be incorporated into the current business process model and consume additional human resources. However, this is the only way how to provide the best accuracy, even if the process of adding this type of identifier would take some time. Even if this type of identifier was rejected, it should be taken into account in the further development of the project.

Accepted methods of reading the item identifier
Unfortunately, the transition to a fully machine-readable format of the item codes is not possible, because according to the packaging company the negotiations with customers are slow and they tend not to change current item code format, and therefore the only possible way is to adapt the solution to the current state. The packaging company strongly prefers methods of reading the item identifier not requiring the change of the current item identifier. Usage of the optical character recognition would remove the need for changing the item codes which reduces the administrative load, but the quality of recognition needs to be verified to achieve desired error rate. Another advantage of optical recognition of current item identifier is the price. Physical form and item type identifier of the item may retain preserved as they are. Selection of this method of item identification raises a question whether the error rate of reading the current identifier would be low enough to fulfill requirements of the packaging company.

2.2.2 Scanning device

There are several devices designed for optical recognition currently available on the market. Candidates are described below, despite the fact that not all of them fulfill the industrial grade standards.

2.2.2.1 Industrial OCR camera

The first type of devices designed for scanning and OCR is the industrial grade reader. This device consists of a camera and embedded computer in a durable enclosure with ingress protection rating up to IP68. It offers high quality of reading the codes thanks to quality optics with auto-focus and additional lighting. Unfortunately, those devices are usually not intended to be hand-held. They are designed to be mounted on a frame in a static position which is not desired in this project because of various locations of the item identifiers on the items. Their primary use-case is the scanning of identifiers on items traveling on conveyor belts. Those devices also require a computer with a display device to perform the verification according to the packaging instructions and guide the human operator through the process of item types verification using user interface. In this case, all parts of the scanning device would be located at each packaging line. If we want to preserve the ingress protection as the OCR camera has, other components of the reading device should offer the similar IP rating (e.g., computer, the display, the user input device). This setup allows logging of the packaging process on the computer itself, but according to the requirements, there is currently local area network available at the hall with packaging lines. Therefore the data would have to be collected manually by some employee between work shifts, which consumes additional human resources by requiring implementation of another process to the current business process model. The price of this scanning device for one packaging line consisting of the industrial camera, computer and touchscreen is higher than required by packaging company, because just the camera itself cost around 2300 GBP.

SICK Lector 62x[31] is the industrial OCR camera from the SICK company. It offers relatively a small resolution of the image sensor (752x480 pixels) when compared to the current standards. On the other hand, it offers a high level of durability and solid mean time between failures (MTBF) of 75 thousand hours. It offers a large variety of interfaces that could be used for connection with the computer located at the packaging line. The disadvantage is that OCR engine cannot be easily upgraded and therefore keep pace with the current trends in optical character recognition.

Figure 2.2: SICK Lector 62x

Datalogic IMPACT+ OCR[5] is the another suitable OCR camera. It offers higher image sensor resolution (1280x1024 pixels) when compared with the previously mentioned camera. Next advantage is the interchangeability of lens, illuminators, and filters. The enclosure is IP67 rated offering the resistance against dust, water, and oil. List of available interfaces is significantly shorter than in the previous case, but yet sufficient for connection to the computer. Provided software allows easy configuration of the camera and the scanning itself, but the current disposition of item code identifiers is not stable so the location of the identifier can be different for each item type.

Pros

- (+) quality of recognition
- (+) quality of optics (autofocus, interchangeable lenses...)
- (+) durability
- (+) recognition speed

Figure 2.3: Datalogic IMPACT+ OCR

Cons

(-) not designed for hand-held use

(-) OCR engine cannot be easily upgraded

(-) need of additional hardware for managing the scanning process (verification, user interface, input device)

(-) logs need to be collected manually, due to unavailability of the network connection

(-) expensive

2.2.2.2 Hand-held scanners

The design of the solution using hand-held scanners is similar to the previously mentioned setup with OCR cameras. The scanner itself would be connected to the computer taking care of the item type verification and managing the user interface and user input. The advantage is that hand-held scanners allow human operators to scan the item type codes from various angles and positions. This is more suitable for the varying location of item codes. Unfortunately, the scanning conditions are more difficult because of variable focus distance, illumination, angle, etc. The scanning device itself is exposed to worse working conditions because the shock, vibrations or dust are more likely to occur when the device is hand-held.

Camera scanner is the first type of hand-held scanner. The working principle is shared with the industrial OCR cameras with the only difference that the camera is designed for hand-held use and therefore it shares the advantages and disadvantages. The additional disadvantages are that the hand-held scanner is more likely to get exposed to shocks and vibrations and in the most cases the autofocus is not present.

Honeywell Xenon 1900[18] is the example of the hand-held OCR scanner with an image sensor offering resolution 838x640 pixels. The OCR processing is done inside the scanner, and therefore the OCR abilities are limited, and just a limited number of fonts is supported. Concerning the durability and ingress protection, the IP41 is fulfilled, and the scanner should withstand drops to concrete from 1.8 meters.

Figure 2.4: Honeywell Xenon 1900

Line scanner is the device designed to be dragged across the surface of the scanned item by hand. The image is scanned line by line as the scanner is dragged from the start of the item type identifier. The scanning process requires a steady hand because the uneven scanning rate produces the distorted images. The common disadvantage of those devices is that they are not fully intended for industrial use, and therefore the ingress protection is missing.

IRISPen Air 7[21] is the example of OCR scanning pen with built-in optical character recognition. It weighs 28 grams which allow easy manipulation by the human operator. It supports more than 130 languages.

IRIScan Book 5[20] is another line scanner having a much larger width of the scanning line. There is also a difference in processing the recognition of characters. In this case, the characters are processed in a computer to which is the scanner attached.

Pros

(+) mobility

(+) price

Figure 2.5: IRISPen Air 7

Cons

(-) durability and IP rating of the scanner

(-) quality and upgradability of OCR engine

(-) need of additional hardware for managing the scanning process (verification, user interface, input device)

(-) logs needs to be collected manually, due to unavailability of the network connection

2.2.2.3 Handheld personal computer

Considering the disadvantages of the previously proposed solutions the usage of the handheld computer such as smartphone should provide a more suitable solution for this project fulfilling requirements. Thanks to the availability of the rugged phones, having the specifications such as IP68[1] and MIL-STD-810G[6], on the market the durability requirement should be satisfied. Currently available phones also offer a large variety of network interfaces such as cellular network, WiFi or Bluetooth allowing the backup of the logs. They can operate without power cord which could be an advantage as well as disadvantage, because it allows easier manipulation with the device, but on the other hand the battery life has to be longer than one work shift. Those devices provide displaying device and touchscreen for user input in one package which is an advantage when compared to previously mentioned solutions.

Caterpillar **Cat S41** is an average example of the rugged smartphone currently available on the Czech market for approximately 9000 CZK. There are also competitive devices from another companies such as Kyocera[2] and successors as **Cat S61** offering better performance, but also having unnecessary features such as FLIR thermal camera and therefore higher price. The Cat S41 is equipped with Android 7.0 operating system and has octa-core CPU reaching frequencies up to 2.3GHz. The RAM size 3GB and there is a Mali-T880MP2 graphics chip present.

Next important aspect for achieving the high quality of OCR recognition is the quality of camera lenses and image sensor as well as other camera features. The S41 offers 13-megapixel camera equipped with phase detection autofocus (PDAF[10]) allowing faster and more precise camera focus.

Figure 2.6: Caterpillar Cat S41

Pros

(+) durability (IP68, MIL-STD-810G, etc.)

(+) price

(+) cordless - easier manipulation

(+) interchangeability of OCR engines

(+) display and user input device in one package

Cons

(-) cordless - the need for recharging

(-) not primarily intended for industrial use

2.2.2.4 Selection of scanning device

All pros and cons of the mentioned OCR scanning devices were compared and discussed with the packaging company. They would like to achieve high accuracy of OCR, and therefore the use of hand-held scanners was rejected. Considering the complexity of each solution, the setup consisting of the scanning device with computer and display and some kind of user input device is also not entirely desirable. Having all parts of the solution packed in one device lowers the complexity of the whole setup. The solution using the rugged smartphone was preferred mainly thanks to its modifiability and possibility to use latest technologies for OCR, combined with the high-quality camera and the possibility to easily scan item identifier on various locations of the item.

According to previous analysis, the scanning device is supposed to be the rugged smartphone with a custom-made software for OCR and code verification. Usage of the rugged smartphone is cheaper than developing custom hardware with similar processing capabilities and durability. The mobility of the smartphone allows easier use when verifying item codes delivered to the packing line, which would not be possible with the stationary device requiring a constant power supply from an external source. The issue with limited connectivity and the need of recharging the devices will be solved by establishing a defined area within the factory where the chargers, as well as WiFi AP, will be installed, so the smartphones will be able to synchronize the logs with the server during their charging.

There are two major concerns about this solution. First of them is the battery life and whether the scanning device lasts at least one work shift. The second concern is about the quality of OCR when reading current item codes. To be able to verify whether the concerns are justified, a testing device (Cat S41) was purchased tested.

Battery life

The Cat S41 offers a battery with a capacity of 5000 mAh promising long battery life. To measure battery life the production usage had to be simulated. There are three main components with large power consumption: the CPU, screen, and the camera sensor. The screen saver was disabled, and screen brightness was set to 60%, screen was dimming automatically to 30% of brightness after two minutes of inactivity. The phone was put into the airplane mode to eliminate the influence of wireless interfaces to the battery life. The camera application was launched, and the fully charged phone was left displaying the image from the camera. Remaining battery level was checked each hour, and after 10 hours the battery was discharged. The result of this test was found sufficient when we consider the work shift length of 8 hours and the fact that the screen will be turned off when the device is not in use.

Optical character recognition

There are two widespread libraries available for processing OCR on the Android operating system. To verify the quality of reading the item identification codes both were tested.

First of them is the **Tesseract** which is written in C and C++ and is available under the Apache License. The development is sponsored by Google since 2006. This versatile library has a wide variety of settings, but the setup on a mobile platform (Android OS) is quite complicated.

The second library is the **Google Mobile Vision API** offering a complete package for recognizing faces, text, barcodes and QR codes. It is available for Android and Apple iOS. Because it is primarily intended to be used on mobile devices, the setup was much easier. It does not offer such variety of settings but delivers consistently better results in unstructured scenes.

Evaluation set of 100 images of item type codes currently used in production was created, and both libraries were tested in conditions similar to the conditions at packaging line. Each image was manually annotated to indicate what is the correct result of detection. The recognition of code was considered successful if the results of three succeeding detections from OCR engine contained code equal to manually annotated code. All strings were stripped from whitespace characters.

	Tesseract	**Google Vision API**
Accuracy	81%	98%

Table 2.1: OCR library comparison

As the result table 2.1 shows, the Google Mobile Vision API achieved better results. The biggest difference in recognition quality between two tested libraries occurred when the contrast between text and its background was low or when the text background contained various colors (e.g., text placed on color gradient or image). According to the test results and the quality of documentation and ease of setup the **Google Mobile Vision API** was chosen as a library for OCR processing.

2.2.2.5 Conclusion

Various methods of storing the item identifier and its reading were analyzed with the focus to extend the current solution with minimal requirements to the packing company and their customers. The optical character recognition was chosen as a method of reading the current item identifier while keeping additional administrative load close to zero. The recognition itself will be processed on a rugged smartphone as it provides the platform for code scanning, recognition, user feedback and provides a large variety of network interfaces in one package.

The Caterpillar Cat S41 was selected as it fulfills the requirements concerning the durability and battery life. The test of the accuracy of OCR engines provided a suitable candidate meeting the requirements. The usage of a smart phone as a scanning device has additional benefits concerning the price and overall variability of the solution allowing future modifications.

2.2.3 Packaging instructions

Packaging instructions is the document currently used to describe contents of the package, which is generated by the current proprietary information system. To be able to perform automatized verification of the item type identifiers this document has to be provided in a machine-readable format. Thanks to the versatility of the smartphone which was selected as a scanning device and also the fact that the network connection is not available at the packing machine itself, the verification process against packaging instructions should run on the smartphone. Considered ways of delivering the packaging instructions to the scanning device are analyzed below.

2.2.3.1 Suitable methods of data transfer

Transmission over network The packaging instructions will be delivered to the scanning device entirely using the network connection. This method allows fast transfers of a large amount of data. Unfortunately, according to the project constraints the network connection is not available all the time and therefore the transmission to the scanning device would only be possible in previously mentioned charging zone, which complicates the current process. This complication would be removed by establishing a wireless connection across all factory. Transmission over network eliminates the need for the current paper document, which could also be a disadvantage when a fall-back is required.

RFID storage In this case the RFID chip would be used to store packaging instructions in a machine-readable form. This relatively cheap memory storage offering memory sizes up to 4 kilobytes would be attached to the current packing instructions and could be reused after the specific packaging instructions document is no longer valid. Packaging instructions could be transferred to the scanning device as a verification input because there is a support of reading those memories using selected scanning device. Despite the fact that the price is relatively low (0.12 USD/piece) it rises the current costs. This method requires an extension of the current information system and an employee equipped with special device for writing the instructions into the RFID memory, which also brings an additional administrative load.

Optically readable representation After an analysis of currently used information system the possibility to attach current instructions in an optically readable representation to the current packing instructions was discovered. The representation would be printed on the paper together with current packing instructions in the form of machine-readable code such as barcode or QR code. The question is whether the available code types can store sufficient amount of data and whether the scanning device can read this representation regarding the information density, size on the paper, etc.

The use of optically readable code allows seamless integration with the current information system, but it needs to be tested whether it is sufficient representation of data, fulfilling the requirements. There are several code types available in the information system, but a majority of them were rejected with one exception. All of the rejected codes were one dimensional, which are able to store a smaller amount of data per unit size. The exception was the only one two dimensional code available - the QR code[7].

According to the requirements of the packaging company, the maximum amount of UTF-8 characters to be stored in a QR code is around three thousand. Despite the fact this number is theoretical and barely reached in a production deployment, the code containing such number of random characters was generated and printed in size 6x6 centimeters. The printed code was then scanned using the selected smartphone and Google Mobile Vision API in various light conditions containing low light scenarios. The results were satisfactory. In every attempt, the scanning device was able to receive input of given format (QR code) and given maximal size. The disadvantage was that with worse light conditions the reading time was higher.

2.2.3.2 Conclusion

Three methods of storing packaging instructions in a machine-readable form were analyzed. The transmission using QR code does not require any implementation changes in the current information system. The ability to store and read required amount of data was tested with the satisfying result.

2.2.4 Analytics and reports

Whole logging process has several deficits because the logs are currently created manually on the paper. Analysis and performance reports are now done rarely and mainly when something gets wrong, and an error happens. Current state prevents effective work with logs because they are often missing and if not, their processing and analyses are very slow. Mainly because of the duration of manual processing, which includes entering them manually into the computer, which is especially very time-consuming. This duplication of work causes waste of human resources. As the electronic item code verification and logging is a preventive measure, the analysis of the human operator performance is also desired.

2.2.4.1 Data collection

Automatic logging of the whole process by the scanning device could improve the described situation. This feature tries to deal with the human factor, which is in this case insufficient will to create complete and readable logs. Due to unavailability of the network connection, the logs have to be collected offline and temporarily stored without the network connection. This could be done by utilizing the chosen scanning device thanks to its versatility. Synchronization with central storage will occur later in the previously described charging and synchronization area.

To be able to provide complete logs system should support basic operations with following entities:

- human operator
- packaging machine
- item type
- packaging instructions
- log

Requested ability to support multiple lines and even factories, while maintaining high availability of the solution could be hardly satisfied when the on-premise server would be used. To satisfy this request, the horizontal dynamic scaling[32] of server nodes should be used. In that case, the dynamic scaling[22] of the solution is achieved by adding or removing computing nodes instead of improving the performance of one single node. This setup also offers redundancy of compute nodes, preventing solution unavailability in a case of failure of some nodes at the cost of higher management complexity.

Representational state transfer (REST) architectural style will be used. It allows scalable and modular architecture. Server nodes will provide a RESTful API which will be used by clients, in this case, scanning devices, and it will also be used during integration with the current information system.

Data stored and transferred between various parts of the system must be **handled in a secure way** to prevent data leakage and abuse of sensitive data. Considering the nature of the handled data the standard **Transport Level Security** protocol is sufficient and easily implementable.

2.2.4.2 Database engine

The system is required to be capable of collecting and storing data and logs. To fulfill this request, some database system is needed. There are multiple types of database engines available, and each of them has different benefits. To find a suitable candidate, the following must be taken into account.

- write operations will occur with much higher frequency than read operations
- capability of storing large amounts of data for a long time
- distributed nature of the system
- compatibility with analytic tools

Databases can be divided into two major groups SQL and NoSQL. Several aspects of both types of database storages are compared in table 2.2.

	SQL	**NoSQL**
Type	relational	non-relational
Data	structured data stored in tables	un-structured
Table join	supported	not-supported
Schema	static	dynamic
Scalability	vertical	horizontal
OLTP	recommended and best suited for OLTP systems	not recommended due to eventual consistency
Flexibility	rigid schema bound to relationship	non-rigid schema and flexible
Transactions	ACID[17]	CAP theorem[15]
Elasticity	requires downtime	without outage

Table 2.2: Database type comparison

Selection of the database type
Both of the mentioned database types are usable in this solution, but each of them has some disadvantages. NoSQL databases are primarily designed for storing large volumes of unstructured data, which is not the case of this system. On the other hand, much higher level of horizontal scalability is offered. According to the CAP theorem, first published by Eric Brewer, it is impossible to simultaneously provide more than two of following properties: Consistency, Availability and Partition tolerance. Therefore they are not suitable for solutions where read consistency as is critical as in our case.

The SQL databases such as MySQL or PostgreSQL are not that easily scalable, but they guarantee that set of following properties (ACID) is provided.

- Atomicity - all or nothing, if one part transaction fails, then entire transaction fails
- Consistency - any transaction will bring the database from one valid state to another valid state
- Isolation - it is guaranteed that concurrent execution of multiple transactions will result in a state that would be obtained as if they were executed sequentially
- Durability - once the transaction has been committed, it will remain so, even in the event of power loss, crashes or error

A strong argument for choosing the structured database is the structured nature of the data to be stored as well as the guaranteed consistency at all times. Memory requirements of the solution can be estimated by calculating the size of the entity that will occur the most. This entity is the *log* from the scanning process. The sizes of PostgreSQL database columns were used for estimation, and the sizes of the data types required for each property are shown in table 2.3. This estimation gives us a size of the *log entry* of 22 kilobytes.

time	datetime	8 bytes
person	integer	4 bytes
packing line	integer	4 bytes
packing instructions	integer	4 bytes
item type	integer	4 bytes
SUM		22 bytes

Table 2.3: Sizes of the *log entry* properties in a PostgreSQL database

To verify whether the memory requirements will be satisfied the estimated size was compared with size limits of the PostgreSQL (shown in the table 2.4). This rough calculation gives us the possibility to store approximately 1.5 trillion records, which is more than enough to store logs from past ten years.

The result of comparing both types of database storage is that both would be usable, but the SQL offers desired properties such as guaranteed consistency and better suits the nature of the stored data, which is exclusively structured.

Table	32 terabytes
Row	400 gigabytes
Field	1 gigabyte
Number of table rows	unlimited
Number of table columns	250-1600 (depending on column type)
Number of table indexes	unlimited

Table 2.4: Memory limits of example SQL database (PostgreSQL)

2.2.4.3 Analytic tools

When the collected data are stored in the database, it can be processed to provide business intelligence data. Custom implementation within the server software can do the processing, or some kind of analytic tool can be used. The usage of the analytic tool is preferred due to its versatility and extensibility because it supports creating new metrics in a relatively simple way.

The analytic tool can be connected directly to the database with logs and create reports and charts using only read operations, and therefore the persisted data are not modified in any way. There are several tools available, and they are listed below:

- Datadog
- Kibana
- Grafana
- freeboard

Because their features are competitive and none of them protrudes, the Grafana was selected because it is a well documented, free open-source software. The connection to the selected PostgreSQL database is supported without any modifications. Grafana supports various types of data visualizations such as charts and tables and therefore should be sufficient to meet the requirements.

2.2.5 User interface

The user interface is required to guide the human operator through the item type verification process. There are special requirements on the user interface because the needs and capabilities of the target group (human operators) are specific. They need to analyzed to be able to provide an intuitive user interface. Operators have zero to minimal experience with smartphones, and therefore it is needed to be aware of this fact when designing the user interface. To overcome this issue together with reducing the cognitive load it is required to minimize the interaction with the device. The characteristics of the target group are listed below:

- working in industrial environments

- work shifts are long and sometimes night
- experience with consumer electronics is close to zero
- more males than females
- education is low
- low discipline to create complete logs
- without knowledge of foreign languages (except Czech)

During the design of the user interface for this specific target group, following aspects have to be taken into account:

- easy to understand
- visible status of the system
- minimal interaction required
- low number of control elements
- sufficient size of the control elements
- avoid complex operations
- avoid words from foreign language and difficult terms

Selected scanning device will provide the user interface as it is capable of displaying the interface to the human operator as well as read inputs from the operator using the touch-screen. Transformation to corporate-owned, single-use device (COSU[8]) allows reduction of device capabilities and therefore lowering the complexity of using this device while preventing from unintended use.

The some of the characteristics of the target group are corresponding with characteristics and requirements of elderly users. The experience with electronic devices is low. The learning ability is lower in comparison with an average adult. The cognitive ability during overnight work shifts is lowered as in case of elderly. This similarity was used by applying some of the guidelines of user interface design for elderly.

According to the Rodrigues et al. [28], the weakness of one modality can be counterbalanced by combining multiple modalities, resulting in increased usability and accessibility. To achieve similar results the maximum of modalities provided by the scanning device should be used. The scanning device provides modalities such as:

- visual
- acoustical
- haptic

The user interface will be designed using the principles of user-centered design (UCD[11]) as the capabilities of the target group together with ease of use are crucial. It should require minimal interaction from the side of the human operator. The system status should be visible at all time at the feedback to the user should be provided using multiple modalities. The control elements should be up-sized, and the control should not require any sophisticated gesturesSchneiderman [29]. The next heuristic that will be used for UI design has its roots in the predictive model called Fitt's law[30]. To minimize the time and effort needed for navigation within the user interface the control elements should be located close together and have sufficient size.

The user interface needs to be evaluated whether the previously mentioned aspects were taken into account. The basic usability inspection is the heuristic evaluation[26], followed by cognitive walkthrough[27]. Usability testing will be used to evaluate the user interface by testing it on human operators. Several participants will be selected, and qualitative testing of the UI will be done.

The COSU mode prevents users to use the device in an unwanted way. Operating the device in the COSU mode allows transformation of the Android smartphone into a single purpose device while maintaining the possibility to extend its functionality in the future.

2.2.6 Communication and connectivity

Issues were found in communication between the human operator at packing line and the distribution center. The logs are collected manually after the work shift, and human operators are requesting new palletes with items using one wired telephone shared between multiple operators.

There is no local area network currently available in the hall with packing lines and there are several suitable technologies capable of providing stable, industrial grade connection such as: **CAN-BUS**, **FlexRay**, **SigFox**, **ZigBee**, **IQRF** and **WiFi**.

It is required that solution has minimal requirements in this aspect and therefore the WiFi area was chosen concerning the capabilities of the selected scanning device. This area requires setting up a WiFi access point to provide a network for synchronization to scanning devices while they are charging.

The further development in this area was strongly recommended and should be taken in care during future project phases.

2.3 Conclusion

Various possibilities of storing and reading the item type identifier were analyzed. The preferred solution is to read current item identifier using the camera of a rugged smartphone and recognize the item code using the Google Mobile Vision API library for OCR recognition. Packing instructions will be transferred to the smartphone using QR code generated by the current information system and printed on paper as it provide backward compatibility. Due to unavailability of network connection, the logs from verification process will be created and temporarily stored locally on the scanning device. The synchronization will occur after the work shift while the device is recharging in the synchronization zone. Scanning devices will communicate with the logging server using RESTful API, and the data will be stored in a relational SQL database. The analytics will be performed using free analytic tool Grafana as it meets the requirements and it is open source. The user interface guiding the user through the verification process will be displayed directly on the scanning device. It will follow guidelines and heuristics for simple and intuitive design, and the evaluation will be performed.

Chapter 3

Design

This chapter is dedicated to the design process and architecture description of the solution.

3.1 Requirements

Main functional requirements of the system are:

- item type verification against packaging instructions document
- logging of the verification process
- analysis of the logs
- intuitive guide for human operators

Diagram 3.1 shows interactions between system and actors. Those actors are management, operators of the packaging lines, and proprietary information system. Individual use cases are briefly described below:

Log into system The human operator of the packaging line is identified. The identity of the operator is used during the logging process.

Load packaging instructions The human operator specifies packaging instructions to be used for verification of the package contents.

Verify item type The human operator verifies whether the item of given type belongs to the current package.

Log verification of item type The verification of the item type is logged into the system.

Browse logs The manager uses the system to browse the logs from verification process.

Perform analyses The manager uses the system to perform analyses using collected logs.

Manage employees, packaging lines, packaging instructions, item types The external proprietary information system or manager uses this system for management of employees, packaging lines, packaging instructions and item types.

Figure 3.1: Use case diagram

3.2 System description

According to the analysis, the solution (fig. 3.4) fulfilling requirements was designed. It consists of scanning devices operated by human operators at the packaging lines. Those scanning devices will be used for recognizing the current item type identifier using OCR technology, guiding the operators through packaging process (shown on figure 1.2) using intuitive user interface and logging the verification process. Due to unavailability of the network, the logs will be temporarily stored locally and then synchronized with the server during battery recharge in a specified area covered with the wireless network connection. Scanning devices communicate with the server using REST API. The server stores information using the relational database. The Grafana, analytic and monitoring tool, running on separate server instance will be used to process analyses.

3.2.1 Entities

Entities occurring in the system and their relations are shown in diagram 3.3. Their properties are described below:

- **User** - User of the system such as operator of the packaging line or manager. Despite unique identifiers, there are also first and last name and role saved. The *user* entity can be author of another entities such as *item type*, *packaging instructions* or *log*.

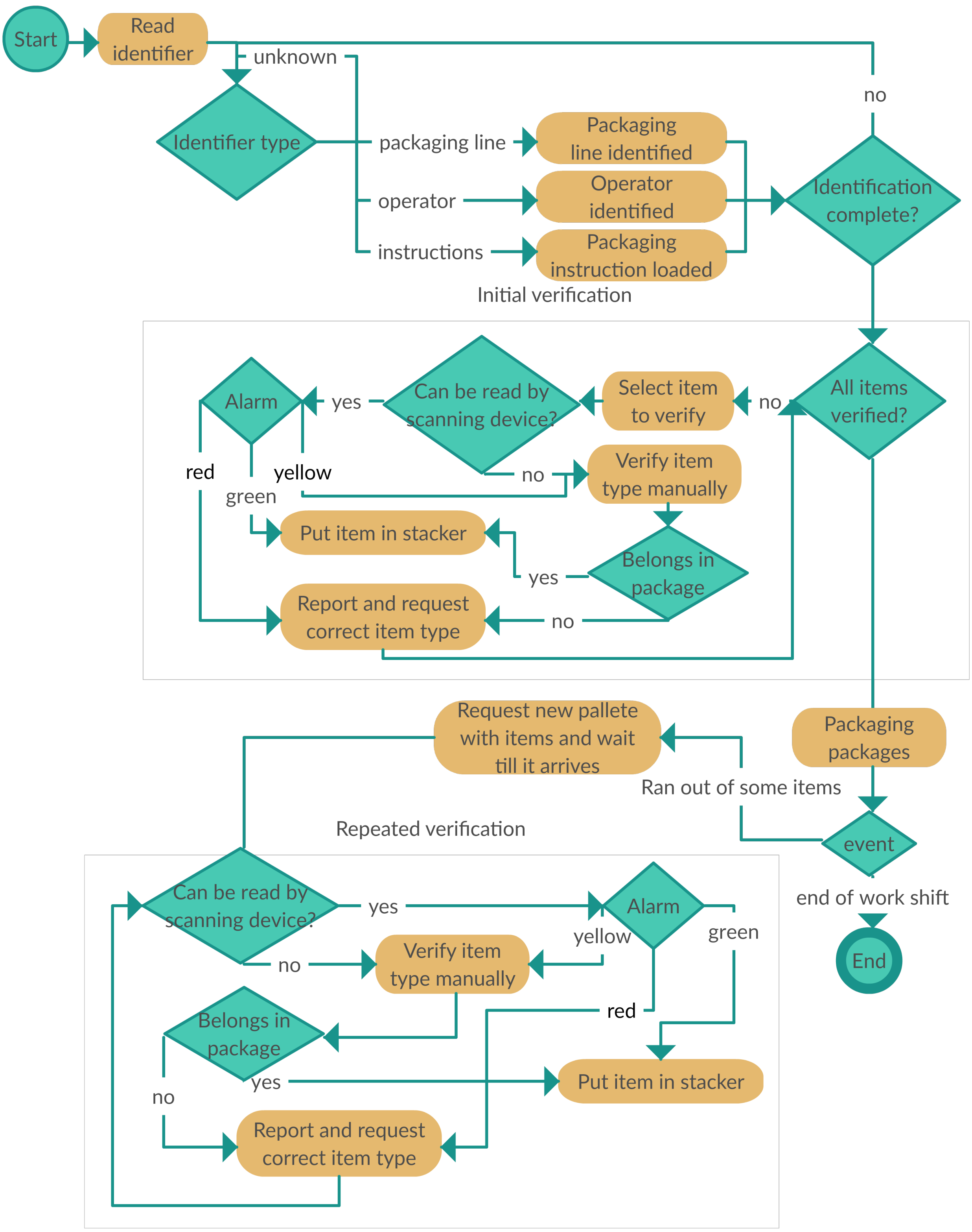

Figure 3.2: New Verification process

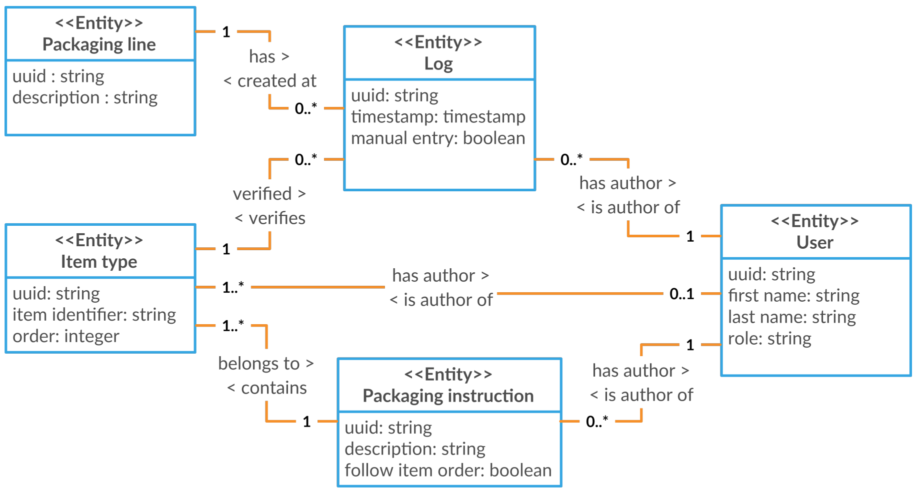

Figure 3.3: Entities and their relationships

- **Packaging line** - This entity represents *packaging line* to distinguish where the *log* was created. It also contains text description (location of the packaging line, identifier within factory, factory etc).
- **Item type** - Representation of item type including its identifier and order within *packaging instructions*.
- **Packaging instruction** - Representation of current packaging instructions containing additional information such as whether the item order should be followed or not.
- **Log** - This entity represents the verification of item type. It has relations to another entities such as *packaging line* (where the log was created), *item type* (which item type was verified) and *user* (who verified the item type). The *log* also has properties noting when the log was created and whether the verification was performed automatically (using OCR) or manually (missing code etc.).

3.2.2 Components

Individual components of the system are described below:

- **Scanning device** - The hand-held personal computer used by the operator of the packaging line. This device is used for verification of the item types using the OCR and logging those verifications.
- **Application server cluster** - Setup of multiple computers providing the API used primarily by *scanning devices* allowing them to store logs into the database.
- **Database server** - Used by nodes from application server cluster to store entities mentioned in the previous section.

- **Analytics server** - Server with analytic tool Grafana used to perform analyses and statistics. It is directly connected to the database server.
- **Client devices** - Devices used by managers to view visualizations and logs from Grafana.

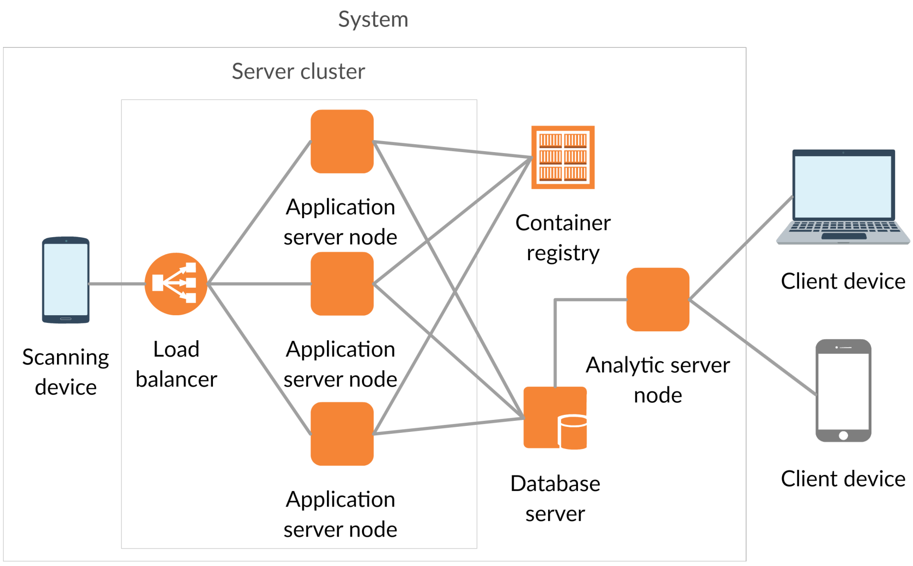

Figure 3.4: System components

3.3 Architecture

The solution utilizes representational state transfer (REST) architectural style[14], which means the devices are divided to clients (such as scanning devices) and server devices providing application programming interface (API) to client devices. Thanks to the statelessness of the API, each one of the server nodes is able to process request from clients.

The architecture of the solution is multi-layered. The first layer is the **data access layer** simplifying the access to data stored in the database. The second layer is the **business layer** enforcing the business rules by which the entities are created and modified. Next layer is the **GRASP controller layer** and the last one is the **presentation layer** taking care of displaying user interface.

3.4 Deployment

Deployment diagram (figure 3.5) models the physical deployment of the solution on computational devices. Those devices are the scanning devices communicating with some of the server nodes from computing cluster. The specific server node is selected by the load balancer, which distributes the requests from clients. The size of the server cluster (count of server nodes) is adjusted dynamically according to the load. When the compute cluster is scaled up a new node is allocated, and the application server is started using operating-system-level virtualization (also known as containerization[9]). The container with application server is created using *image* obtained from *container registry*. There is always one application server running on one cluster node. When the cluster is scaled down, one (or more) of the nodes is stopped, and resources are released. This deployment scheme also reduces the risk of the service unavailability due to an error in one of the nodes. In such case, the requests are rerouted to healthy instances.

3.4.1 Deployment

To lower the complexity of solution maintenance the products from Amazon Web Services (AWS) are used for server part of the solution.

- **Application server cluster** - Amazon EC2 Container Service (ECS)
- **Container registry** - Amazon EC2 Container Registry (ECR)
- **Database server** - Amazon Relational Database Service
- **Analytic server node** - Amazon EC2 Instance

3.5 Communication

This section describes the communication between system components.

3.5.1 Application server API

The RESTful API is provided by application server nodes. This API is used by clients such as scanning devices or current proprietary information system to interact with the system. Description of the API using OpenAPI specification is attached in the appendix. The Swagger tool was used to create this description.

3.5.2 Other

Other technologies for communication were used. Each of the server nodes is connected to the database using JDBC unifying the access from Java code across various types of SQL databases.

3.6 User interface

This section is dedicated to the design process of the user interface for operators of the packaging lines.

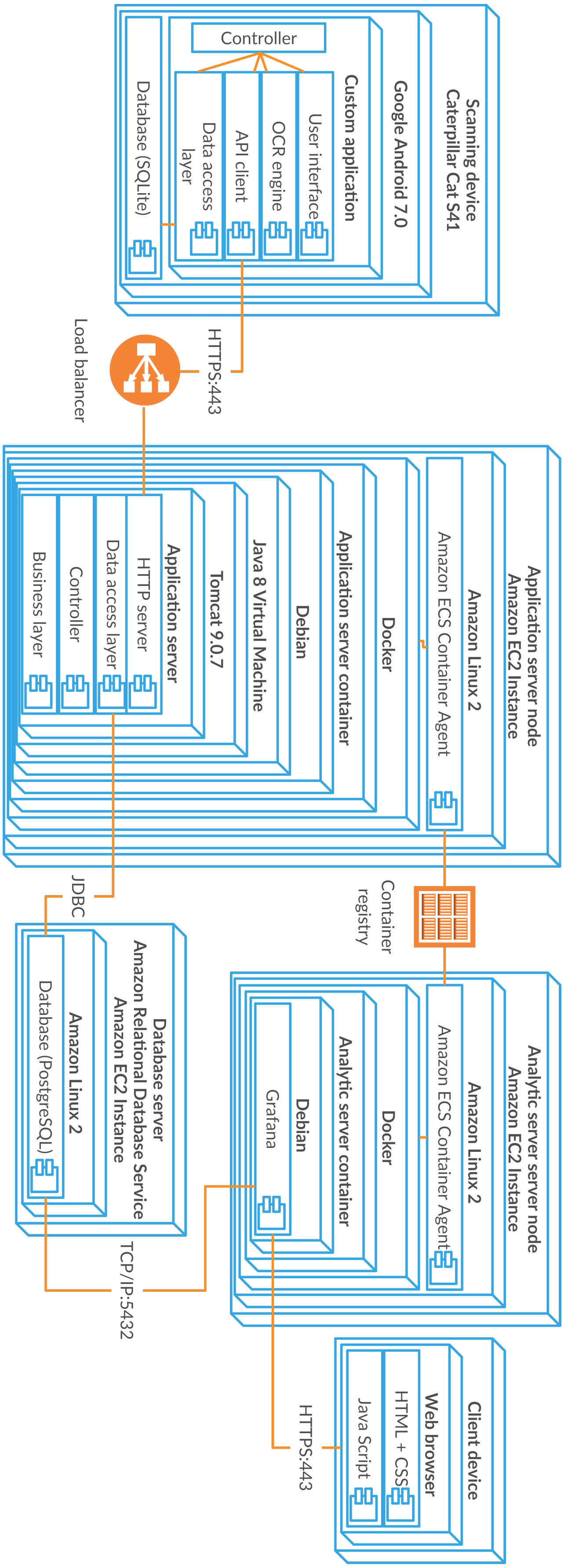

Figure 3.5: Deployment scheme

3.6.1 Android design guidelines

The operating system of the selected scanning device is the Google Android 7.0 Nougat. Google released a set of rules and recommendations for developers, called Android Design Guidelines. The goal of those design guidelines is to provide a consistent look and feel to the Android users across all applications. The visual aspect of the application is described by design language called **Material Design**. Three main principles of the Material Design are listed below:

- **Material is the metaphor** - Light, shadows and movements are used. The display should be taken as a real-world three-dimensional environment.
- **Bold, graphic, intentional** - Immediately apparent functionality and waypoints for the user.
- **Motion provides meaning** - Usage of motion to focusing attention and maintaining continuity.

3.6.2 Design process

The primary purpose of the user interface is to guide the human operator through the packaging process. According to the analysis, special requirements of the target group were taken into account. Main aspects taken into account during interface design are listed below:

- **System status visibility** - The system status should be clear to the user at all times.
- **Large control elements** - Control elements should not require any precise or complex commands.
- **Minimize interaction** - Minimize interaction from the user and perform as many actions as possible automatically to lower the cognitive load.
- **Multi modality** - User feedback uses multiple modalities to improve the perception of the system state.

A system of *alarms*, as a basic form of feedback, was created to help operator with understanding the system status. There are multiple alarm types with various color codes and meanings. All alarms except the gray one are a possible result of the item verification. Meanings of all alarms are further described below:

- **Gray alarm** - Informs the user about the system status. No action is required.
- **Green alarm** - Shown when the item type verification against packaging instructions was successful and therefore the item of given type belongs to the package.
- **Yellow alarm** - This alarm states that the code is likely to belong into the package, but the verification could not be done automatically. This can be caused be the insufficient accuracy of the OCR engine. The differing characters or characters with low confidence are marked to help with manual verification of the code. Simple user interaction is needed afterward.
- **Red alarm** - The scanned item type does not belong to the packaging instructions.

3.6.3 Prototyping

According to the analysis and requirements of the target group two user interfaces were designed using paper mock-up technique. The well-known heuristic evaluation[26] method developed by Jacob Nielson and released in 1994 was used during design.

3.6.3.1 Design 1

The user begins at the screen displaying the full-screen preview from the camera. He is prompted to scan all necessary identifiers such as user identifier, packaging machine identifier, etc. Progress is shown at the bottom part of the screen. When all of the necessary identifiers are scanned, the verification process begins. The verification process is started by scanning the item identifier and the result of the verification shown using previously described system of alarms. In a case of yellow alarm, the list of the package contents is scrolled to the specific item and item is automatically moved left, unveiling manual override buttons.

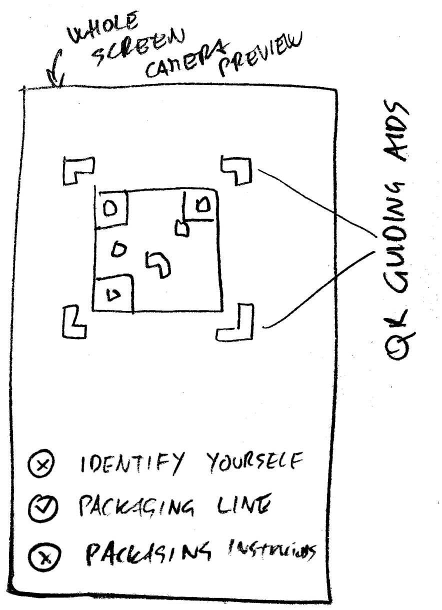

Figure 3.6: Design 1: Initial screen for scanning identifiers of human operator, packaging line and packaging instructions

3.6.3.2 Design 2

The second design is landscape oriented. This should allow easier aiming to the item identifier code because of the horizontal orientation of identifiers. The process is started by bringing the user to the overview screen where scanning the identifier of a specific entity (such as operator, packaging line, etc.) can be invoked. After all of the required entities are scanned, the full-screen camera preview is displayed. The verification process is started by scanning the item identifier and the result of the verification shown using previously

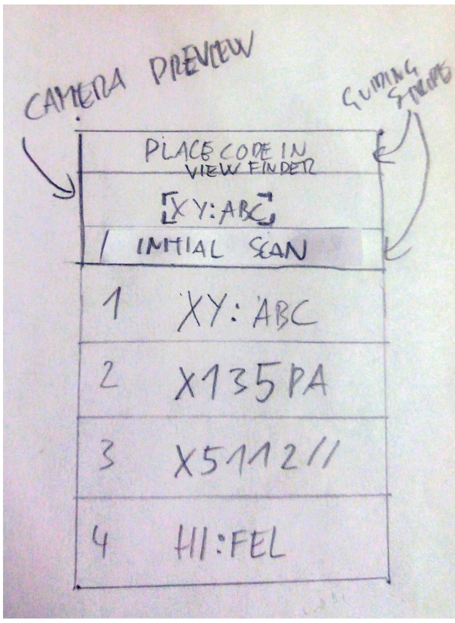

Figure 3.7: Design 1: Verification of item type code

described system of alarms. In a case of yellow alarm, the list of the package contents slides from the right and the modal confirmation window is shown.

3.6.4 Evaluation and comparison of the user interfaces

Usability inspection without real users allows faster prototyping in early phases. Instead of testing the user interfaces with real users an expert reviews the user interface design, using usability inspection methods such as heuristic evaluation[26] and cognitive walkthrough. The expert review of both prototypes was performed, and the findings from inspection were compared to perform the selection of one candidate.

3.6.4.1 Design 1 findings

There were no major issues found in this user interface design. It provides visible feedback, and the user is guided in every step of the verification process. It offers a smaller preview of the camera image, but on the other hand, the list of item types is visible at all times providing feedback on the verification process.

Hidden options The largest weakness of this design is the unclear way of manual verification of the item type. In a case of yellow alarm the row is swiped left automatically, but in a case of manual verification, it may not be clear what action should follow.

3.6.4.2 Design 2 findings

This design is a legit candidate, but several design flaws were localized during testing. To be able to display a larger preview of the camera, the landscape orientation is used. This

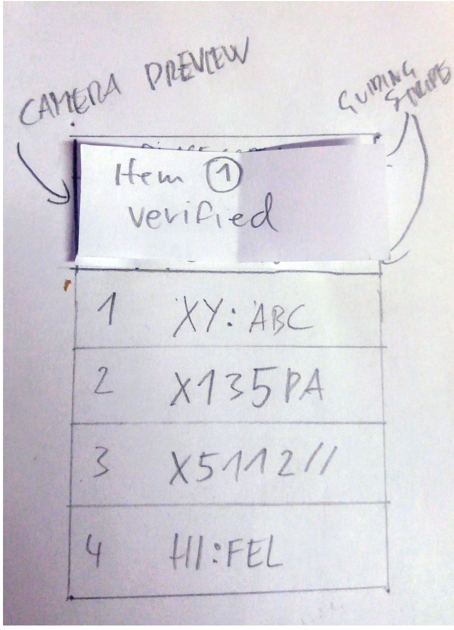

Figure 3.8: Design 1: Visual feedback

should help with the aiming of the camera, despite the fact, the list of item type is hidden by default.

Screen navigation There are multiple unnecessary transitions between screen creating the possibility of the user losing track of system status. This occurs especially when the user identity, packaging instructions, etc. are scanned.

Hidden options The list of item types is hidden by default, making the progress of the process invisible to the user. Displaying the list requires an additional swipe. The dialog with manual verification option is invoked by pressing the item box, causing the fragmentation of visual aspect of the user interface, causing the possible increase in cognitive load.

Screen orientation Landscape orientation was utilized to maximize the size of camera preview to improve the aiming to the item code. Despite the advantages, the possibility to operate the device one-handed was impeded. The list of items for verification also cannot be displayed due to the landscape orientation. This may cause the user to lose track of the progress of verification process.

3.6.4.3 Conclusion

After the comparison of findings from both designs, the first design was selected. It offers a less fragmented user interface with fewer control elements while allowing better operation in one-handed mode, which allows the operator to use the other hand for item manipulation. The issue with the unclear invocation of manual verification will be solved by tutoring the human operators, as the advantages of showing the whole list prevails.

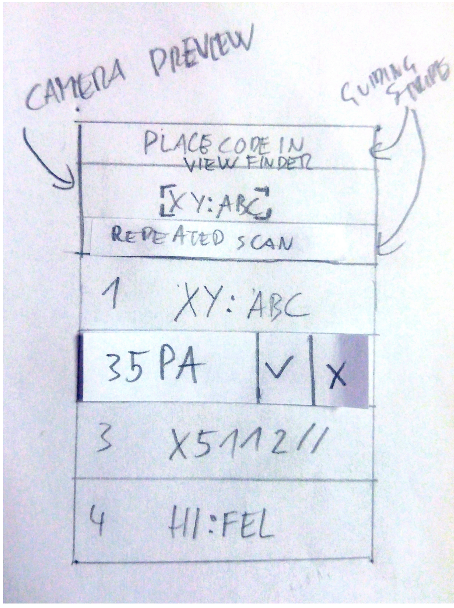

Figure 3.9: Design 1: Manual override of the verification

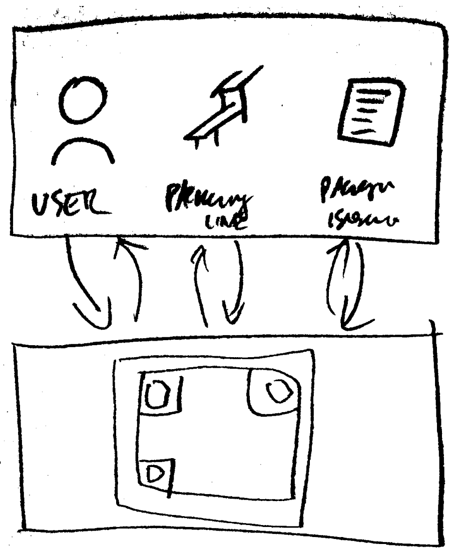

Figure 3.10: Design 2: Initial screen for scanning identifiers of human operator, packaging line and packaging instructions

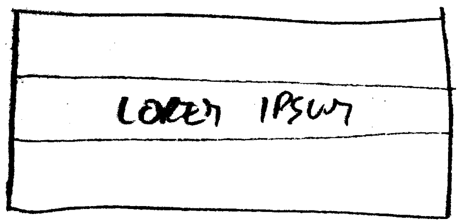

Figure 3.11: Design 2: Verification of item type code

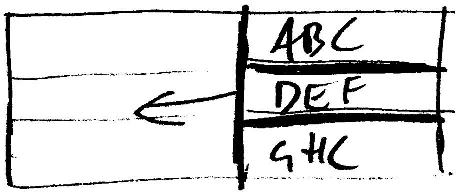

Figure 3.12: Design 2: Content of packaging instructions

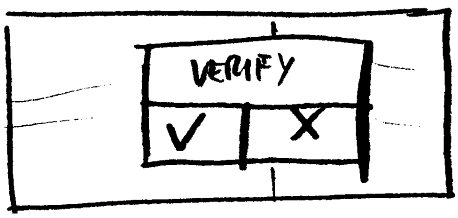

Figure 3.13: Design 2: Manual override of the verification

Chapter 4

Implementation

The solution was implemented according to the design specified in the previous chapter. Tools used for implementations are mentioned in the sections below.

4.1 Scanning device

Parameters of the selected scanning device (Caterpillar Cat S41) are shown in table 4.1.

CPU	Mediatek MT6757 Helio P20 octa-core
GPU instructions	Mali-T880MP2
RAM	3GB DDR3
Storage	32GB
Camera	13MP, PDAF, flash
Operating system	Android 7.0 Nougat

Table 4.1: Parameters of scanning device

Android Studio 3.1.1 (Build 173.4697961) IDE together with Android SDK were used for the development of the scanning device software. The second revision of **Android SDK API level 24** (Nougat) was selected according to the version of the operating system of the scanning device. Java was used as a programming language. Git was used for version control during whole development.

Gradle 4.4 was used for building and testing purposes as well as for dependency management.

According to the analysis the **Google Mobile Vision API** was used for OCR processing. This OCR engine is as part of **Play Services** library. Next group of libraries was used for implementing the user interface, and it contains libraries from Android support package (constraint-layout, design, support-v4, support-v13, appcompat-v7, recyclerview-v7, cardview-v7) as well as a library from third-party developers (Daimajia Swipelayout). There are also two libraries from Apache for string manipulations of item identifiers: Apache Commons Text and Commons Lang 3. The last group contains support libraries used such as **Lombok** or **Google Gson**

Name	Version	License
Android OS	7.0	Apache License 2.0
Android Studio	3.1.1	Apache License 2.0
Android SDK	level 24 r2	Android SDK License
Gradle	4.4	Apache License 2.0
Android Play Services	12.0.1	Android SDK License
Android Support	27.0.2	Android SDK License
Daimajia Swipelayout	1.2.0	MIT License
Apache Commons Text	1.3	Apache License 2.0
Apache Commons Lang 3	3.7	Apache License 2.0
Lombok	1.16.20	MIT License
Google Gson	2.8.4	Apache License 2.0
Java (JDK, JRE)	1.8	GNU GPL
Git	2.17	GNU GPL

Table 4.2: Scanning device: libraries and licenses

Table 4.2 shows version and licenses for each tool or library used during development. All the third-party software is released under licenses suitable for usage in commercial software.

4.2 API server

According to the design phase, the solution runs on AWS EC2 Instance. Despite the fact there are various instance sizes available, the smallest instance size (t2.micro) was used in compute cluster to provide higher granularity in solution scalability. Parameters of this instance are described in table 4.3.

CPU	Intel Xeon (1 CPU)
RAM	1GB
Storage	22GB EBS
Operating system	Amazon Linux 2
Additional software	Docker 17.12.1-ce, Amazon ECS Agent

Table 4.3: AWS EC2 instance parameters (t2.micro)

The application server is written in the Java programming language using IntelliJ IDEA IDE in combination with Maven as a build and dependency management tool. The Git was used for version control during whole development. API server runs as a web application providing RESTful API inside the Apache Tomcat, an open-source Java Servlet Container. Jersey was selected as an implementation of the JSR-339 API, which provides support in creating web services according to the REST architectural pattern. Communication with PostgreSQL database storage is realized using JDBC API. The entities are stored using Hibernate framework, allowing mapping of the object-oriented domain to a relational database.

Table 4.4 shows version and licenses for each tool or library used during development. Tools and libraries are published under various licenses. All of the third party software is

Name	Version	License
Apache Tomcat	9.0.7	Apache License 2.0
Java (JDK, JRE)	1.8	GNU GPL
Glassfish Jersey	2.27	GNU GPL
FasterXML Jackson	2.9.5	Apache License 2.0
Lombok	1.16.20	MIT License
Hibernate ORM	5.3	GNU LGPL
PostgreSQL	10.3	PostgreSQL License
Git	2.17	GNU GPL

Table 4.4: API server: libraries and licenses

published under licenses allowing commercial use.

4.3 User interface design

The development of the user interface went through several phases. After the analysis of the requirements and recommendations, the paper mockups were created. After selection on one of the available candidates, the low fidelity prototype was created using the Balsamiq Web App tool. This low fidelity prototype was inspected using heuristic evaluation method and then implemented as a part of the software for the scanning device.

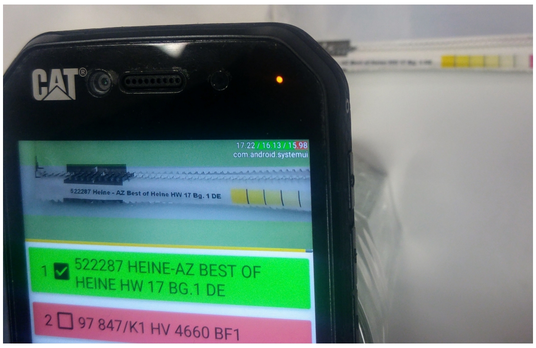

Figure 4.1: Early stage of high fidelity prototype

The user interface was designed according to the requirements. Each screen clearly shows the current status of the system preventing the operator from getting lost. The system always provides immediate visual and acoustic feedback. As mentioned before technical competence of operators is not high and therefore natural language is used without complicated terms. The effort to prevent errors and mistakes utilizing a system of alerts and confirmation dialogues. The transition between states is always enabled to give the operator a possibility to leave the unwanted state. During the user interface design, the Android operating system design guidelines were taken into account to keep the UI intuitive. The transfer of the responsibility from operator to machine is realized via clear visibility of system status which helps the operator to recognize what is the current state of the system rather than recall it.

Design of the system is kept as minimalistic as possible to minimize distraction of operator and to keep important information visible as much as possible. Errors messages are designed to clearly and precisely describe what is happening to help operator recover to the normal state.

Chapter 5

Evaluation

This chapter is dedicated to the evaluation of the implemented solution. The solution was tested in a simulated environment with conditions reflecting the conditions at the packaging line as well as at the packaging line itself. Two main areas that were tested are the quality of identification of the item type using the OCR technology and the user interface for human operators.

5.1 Item type verification

The ability to verify the item type was evaluated on a production data set. The content of the evaluation data set is shown in the table 5.1.

Number of packaging instructions	20
Average number of item types in package	12.05
Number of unique item types	241

Table 5.1: Evaluation dataset

The main aspect influencing the quality of the OCR recognition is the ambient illumination. To accurately simulate the production environment the illumination was measured at the packaging lines with average illumination level of 488 lux. The production environment is equipped with fluorescent tubes mounted on the mount located approximately 4 meters above the floor. The type of illumination, as well as the displacement of the light sources, also need to be considered for accurate simulation of the production environment. The test in the simulated environment was conducted as an internal verification of functionality. The final test in the production environment was conducted with the participation of multiple employees, and the verification itself was performed by the actual human operator of the packaging line.

The evaluation dataset was intentionally corrupted, so it contained items with modified identifiers or items not belonging to the given package. Special effort was dedicated to the real rate of false positives as the most dangerous type of verification error. The possible results of verification are shown and described in table 5.2. The yellow alarm was included in unwanted results as it requires the attention of the human operator to perform the manual

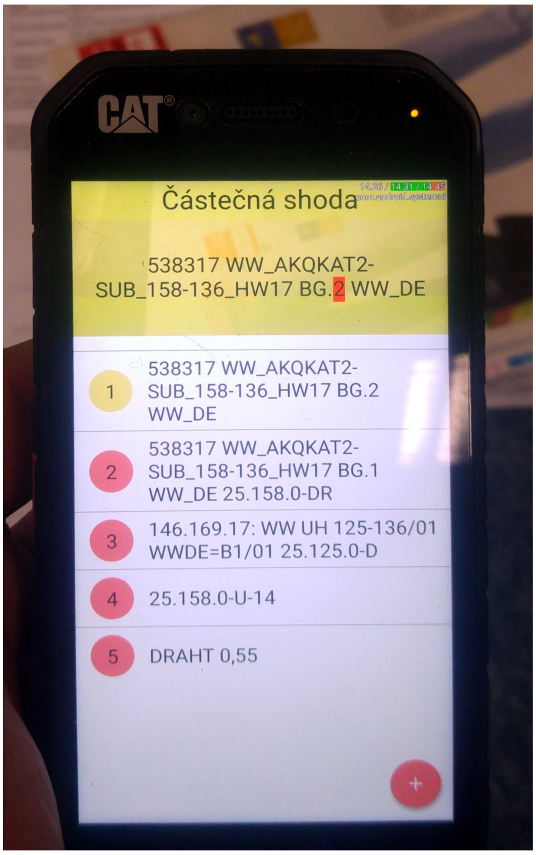

Figure 5.1: Evaluation in simulated environment

verification, despite the fact it helps by highlighting the difference between closest of the available item codes from the packaging instructions. The result of the evaluation in the production environment is shown in the table 5.3.

TYPE	**DESCRIPTION**	**DESIRED**
true positive	The item belongs to the package and the verification results in the green alarm with the correct item type.	YES
true negative	The item does not belong to the package and the verification results in the red alarm.	YES
false negative	The item belongs to the package and the verification results in the red or yellow alarm or the green alarm with different item type.	NO
false positive	The item does not belong to the package and the verification results in the green or yellow alarm.	NO

Table 5.2: Possible verification results

The results of the test show that the accuracy of verification is 97.09%. The false positives occurred five times, meaning that in 2% of cases the scanning device evaluated that the scanned item type does not belong into the package even if it belonged. This is the less

Result type	Item count	Percentage
true positive	209	86.72 %
true negative	25	10.37 %
false negative	5	2.07 %
false positive	2	0.83 %

Table 5.3: Evaluation result

serious type of error because the scanning device serves as and additional equipment and the code should be verified manually. When the yellow alarm occurs the differences are highlighted, and the user experience is improved even if this result was considered as an error. The most serious type of errors are the false positives, meaning that the scanning devices evaluate the item type as belonging to the package even if it is not. This could be improved by replacing the OCR engine with the better one, but even the current result offers error rate lower than the current state.

5.2 User interface testing

The evolutions of the user interface of scanning device were inspected during design phase without real users. The usability testing with users was performed to evaluate the usability of the high fidelity prototype. The qualitative testing using think aloud-method[24] was performed with three participants currently employed as operators of the packaging lines. The participants were interviewed before and after the test. The interview before the test was focused, among other things, on their will to use such system. The goal of the interview after the test was to calculate the system usability scale[4] score.

5.2.1 Description of participants

Table 5.4 shows the description of test participants. Selection of the participants was made in a way, that the participant group is diverse. This helps to fully test various requirements of various members of the target group. Special care was dedicated to select participants diverse in age, education, and experience with electronic devices.

	Gender	Age	Education	Consumer electronic experience	Employment duration
Participant 1	male	32	none	low	5 years
Participant 2	female	23	elementary	low to average	1 year
Participant 3	male	55	elementary	zero	22 years

Table 5.4: Description of participants

5.2.2 Specified task

The test is focused on finding how easy is for new users to accomplish tasks with the system. All participants were requested to walk through whole verification process including the repeated verification. Main steps of task are specified below:

- identify themselves, machine and read packaging instructions
- perform initial verification of item types
- perform repeated verification of item types

Packaging instructions were prepared especially for the usability testing. One of the item types did not have an identifier and therefore required manual verification. One of the items in the repeated verification had modified the code to throw yellow alarm requiring manual verification.

5.2.3 Usability test

The usability test was performed in the production environment at the packaging line to truly simulate normal usage of the system and the conditions such as lighting etc. The process was the same as it was in production, including the manipulation of items, etc. Each of the participants performed the verification of items using the scanning device and was asked to *think aloud* while performing verification. Their thoughts and comments were logged, and the logs are available in the appendix. The facial expressions were logged as well by another member of the evaluator group. After they finished the test, they have been interviewed to calculate the SUS score.

5.2.4 Test results

The test went smoothly, and the SUS score for each participant calculated from the post-test interview as well as their average are shown in table 5.5.

All of the participants agreed that the system lowers the amount of their work. They handled the scanning device surprisingly well, despite the fact that their experience with consumer electronics is very low. Two participants liked the simplicity of the color-coded system of alarms, as it provides immediate and understandable expression of what is happening. All of the participants praised the ability to highlight the difference in scanned code against packaging instructions. They immediately understood the meaning of the highlighted parts of the item code.

5.2.4.1 Critical incidents

The most serious issue happened during the testing session with Participant 3. He could not find a way how to perform the manual verification of the item type, due to his total unfamiliarity with control gestures/elements used by current consumer electronics. He was supposed to uncover the manual verification buttons by swiping the row with given item type

	Part. 1	Part. 2	Part. 3	Avg.
I think that I would like to use this system frequently	3	4	2	3.00
I found the system unnecessarily complex	2	1	3	2.00
I thought the system was easy to use	4	4	2	3.33
I think that I would need the support of a technical person to be able to use this system	2	1	4	2.33
I found the various functions in this system were well integrated	4	4	3	3.66
I thought there was too much inconsistency in this system	1	2	1	1.33
I would imagine that most people would learn to use this system very quickly	3	3	2	2.66
I found the system very cumbersome to use	2	1	3	2.00
I felt very confident using the system	4	3	2	3.00
I needed to learn a lot of things before I could get going with this system	2	1	4	2.33
sum of points	34	37	21	30.66
SUS score	**85**	**92.5**	**52.5**	**76.66**

Table 5.5: SUS score

to the left. Other participants also agreed that this was one of the critical moments. All of the participants would welcome the prolongation of the alarm notification time. Participants 2 and 3 said that it was unclear how to aim camera during item type verification.

5.2.4.2 Conclusion

The participants liked the solution more than dislike it. All of the critical incidents were analyzed and solution provided. The alarm notification time was prolonged, so it provides the user more time to understand the state of the system. The aiming guides were added to the camera preview, so the current item type identifier fits in between. Other methods of invoking the manual verification option were also considered but rejected for not providing sufficient resilience against unwanted invocation. This issue will be solved by tutoring the human operators because the current design of user interface maximizes the area displaying the list of items.

The calculated average SUS score is 76.66, which is almost 9 points above average[3], together with positive feedback indicate that the target group is able to use proposed user interface.

Chapter 6

Conclusion

This thesis dealt with the problem of improving the packaging process and lowering the current error rate. After the identification of the human factor as a major cause of errors, the solution lowering the effect of the human factor, and eliminating weak spots, was proposed. Latest technologies for item identification were analyzed as well as methods to minimize the cognitive load of human operators. The solution was designed concerning the requirements of the packaging company while providing the intuitive user interface for the human operators and preserving the current amount administrative load needed for operation.

The rugged smartphone was selected as the scanning device because it combines the scanning device with input/output devices for human-computer interaction in one package. The OCR technology was selected for item type identification as the result from recognition quality test provided satisfying results. The quality of item type verification was measured using the test in the production environment. The calculated error rate was 2.9% which means that the quality of verification was improved by 9% from current 12%.

The logging server infrastructure was designed after analysis of available technologies. Multiple types of database storages were compared to find to best fit to store logs, packaging instructions, etc. The logging server was implemented using services from AWS portfolio in a way that scalability and stability are ensured. The Grafana was used as a versatile tool for performing the analyses and browse the logs.

The design of the user interface faced a big challenge, because of the target group, with special requirements. Guidelines for user interface design for such target group were analyzed, and multiple design prototypes were created. The proposed user interface was evaluated using both usability inspection and testing methods.

6.1 Future development

The project has potential to help by applying knowledge from multiple areas of computer science into the industry. By making the verification process easier and more automatic, there is an obvious possibility to reduce the error rate resulting in financial benefit. This was one of the first steps in transforming a factory from the 1980s into a smart factory of the 21st century.

Several steps need to be done to complete this transformation into a smart factory. The most important step is the covering factory with the wireless network, allowing real-time communication with scanning devices as well as communication between various parts of the system (requesting new pallete with items, etc.). The collection of verification logs could be done asynchronously (e.g. using queues) to surpass the load peaks. The OCR engine could be improved as well to lower the error rate, especially the occurrence of the false positives as the most dangerous type of error. One of the final steps could be the transformation of the current item code identifier to a fully machine-readable medium.

Bibliography

[1] Degrees of Protection Provided by Enclosures (IP Code). Available from: <https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-IEC-60529.pdf>.

[2] Kyocera Mobile Phone Website. Available from: <http://www.kyoceramobilephone.com/>.

[3] BANGOR, A. – KORTUM, P. – MILLER, J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies*. May 2009, 4, 3, s. 114–123. ISSN 1931-3357. Available from: <http://dl.acm.org/citation.cfm?id=2835587.2835589>.

[4] BROOKE, J. et al. SUS-A quick and dirty usability scale. *Usability evaluation in industry*. 1996, 189, 194, s. 4–7.

[5] DATALOGIC IMPACT+OCR. DATALOGIC IMPACT+OCR. Available from: <http://www.datalogic.com/eng/products/manufacturing-healthcare/vision-systems/impact+ocr-pd-698.html>.

[6] DEFENSE, U. D. O. DEPARTMENT OF DEFENSE TEST METHOD STANDARD: ENVIRONMENTAL ENGINEERING CONSIDERATIONS AND LABORATORY TESTS. Available from: <http://everyspec.com/MIL-STD/MIL-STD-0800-0899/MIL-STD-810G_12306/>.

[7] DENSO. QR Code features. Available from: <https://web.archive.org/web/20130129064920/http://www.qrcode.com/en/qrfeature.html>.

[8] DEVELOPERS, A. E. Android EMM Developers. Available from: <https://developers.google.com/android/work/requirements/cosu>.

[9] DOCKER. docker. Available from: <https://www.docker.com/what-docker>.

[10] EMANUELE MANDELLI, N. K. G. C. Phase-detect autofocus. Available from: <https://patents.google.com/patent/US20160205311A1/en>. WO Patent App. PCT/US2000/009,621.

[11] ENDSLEY, M. R. – BOLTé, B. – JONES, D. G. *Designing for situation awareness: an approach to user-centered design*. London;New York; : Taylor & Francis, 2003. ISBN 9780748409679;074840967X;.

[12] EVANS, D. The internet of things. *How the Next Evolution of the Internet is Changing Everything, Whitepaper, Cisco Internet Business Solutions Group (IBSG)*. 2011.

[13] FEKI, M. A. et al. The Internet of Things: The Next Technological Revolution. *Computer*. February 2013, 46, 2, s. 24–25. ISSN 0018-9162. doi: 10.1109/MC.2013.63. Available from: <http://dx.doi.org/10.1109/MC.2013.63>.

[14] FIELDING, R. T. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.

[15] GILBERT, S. – LYNCH, N. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News*. June 2002, 33, 2, s. 51–59. ISSN 0163-5700. doi: 10.1145/564585.564601. Available from: <http://doi.acm.org/10.1145/564585.564601>.

[16] GRAUMAN, K. – LEIBE, B. *Visual Object Recognition*. Morgan & Claypool Publishers, 1st edition, 2011. ISBN 1598299689, 9781598299687.

[17] HAERDER, T. – REUTER, A. Principles of Transaction-oriented Database Recovery. *ACM Comput. Surv.* December 1983, 15, 4, s. 287–317. ISSN 0360-0300. doi: 10.1145/289.291. Available from: <http://doi.acm.org/10.1145/289.291>.

[18] Honeywell Xenon 1900. Honeywell Xenon 1900. Available from: <https://cdn.barcodesinc.com/themes/barcodesinc/pdf/Honeywell/xenon-1900.pdf>.

[19] HONIG, W. *Operant behavior: areas of research and application*. Century psychology series. Prentice-Hall, 1966. Available from: <https://books.google.cz/books?id=5g5DAAAAIAAJ>.

[20] IRIScan Book 5. IRIScan Book 5. Available from: <http://www.irislink.com/Documents/Image/_IrisLink2.0/Mobile_Scanner/IRIScan_Book_5/idcards/IRIScan_Book_5_Wifi_id_card_EN.pdf>.

[21] IRISPen Air 7. IRISPen Air 7. Available from: <http://www.irislink.com/Documents/Image/_IrisLink2.0/Mobile_Scanner/IRIScan_Book_3/idcards/IPA7-id_card-en.pdf>.

[22] KRZYWDA, J. et al. Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling. *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE*. 2018, 81, s. 114–128.

[23] LASI, H. et al. Industry 4.0. *Business & Information Systems Engineering*. 2014, 6, 4, s. 239.

[24] LEWIS, C. *Using the "thinking Aloud" Method in Cognitive Interface Design*. Research report. IBM T.J. Watson Research Center. Available from: <https://books.google.cz/books?id=F5AKHQAACAAJ>.

[25] MARIK, V. *Prumysl 4.0: vyzva pro Ceskou republiku*. Praha : Management Press, 2016.

[26] NIELSEN, J. Usability Inspection Methods. New York, NY, USA: John Wiley & Sons, Inc., 1994. Heuristic Evaluation, s. 25–62. Available from: <http://dl.acm.org/citation.cfm?id=189200.189209>. ISBN 0-471-01877-5.

[27] NIELSEN, J. *Usability Engineering*. San Francisco, California : Morgan Kaufmann Publishers, October 1994. ISBN 0125184069.

[28] RODRIGUES, É. – CARREIRA, M. – GONÇALVES, D. Developing a multimodal interface for the elderly. *Procedia computer science*. 2014, 27, s. 359–368.

[29] SCHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. Reading : Addison-Wesley, 3rd edition, 1998. ISBN 0201694972;9780201694970;.

[30] SEOW, S. C. Information Theoretic Models of HCI: A Comparison of the Hick-Hyman Law and Fitts' Law. *Human-Computer Interaction*. 2005, 20, 3, s. 315–352.

[31] SICK Lector62x OCR. SICK Lector62x OCR. Available from: <https://www.sick.com/cz/en/identification-solutions/image-based-code-readers/lector62x/icr620c-t11503s50-ocr/p/p352446>.

[32] VAQUERO, L. M. – RODERO-MERINO, L. – BUYYA, R. Dynamically Scaling Applications in the Cloud. *SIGCOMM Comput. Commun. Rev.* January 2011, 41, 1, s. 45–52. ISSN 0146-4833. doi: 10.1145/1925861.1925869. Available from: <http://doi.acm.org/10.1145/1925861.1925869>.

Appendix A

Nomenclature

ACID Atomicity Consistency Isolation Durability

Amazon EC2 Amazon Elastic Compute Cloud

API Application Programming Interface

AWS Amazon Web Services

CAP Consistency, Availability, Partition tolerance

COSU Corporate-Owned, Single-Use

CPU Central Processing Unit

DB Database

GPL General Public License

IDE Integrated Development Environment

IoT Internet of Things

LGPL Lesser General Public License

MTBF Mean Time Between Failures

OCR Optical Character Recognition

OLTP Online Transaction Processing

PDAF Phase Detection Autofocus

REST Representational State Transfer

SQL Structured Query Language

SUS System Usability Scale

UCD User Centered Design

UI User Interface

UX User Experience

Appendix B

Low fidelity prototype

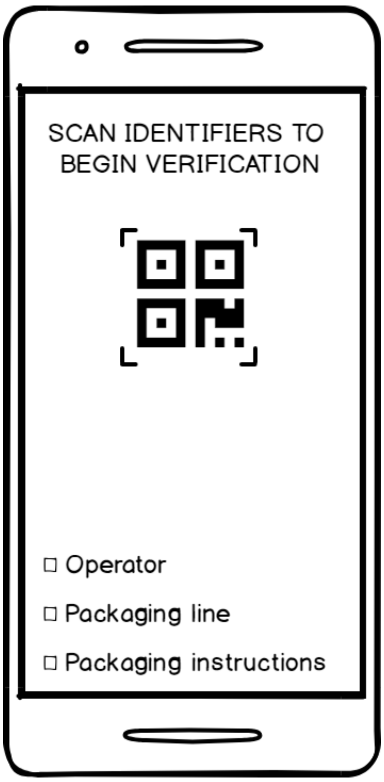

Figure B.1: Scanning device - mock 1

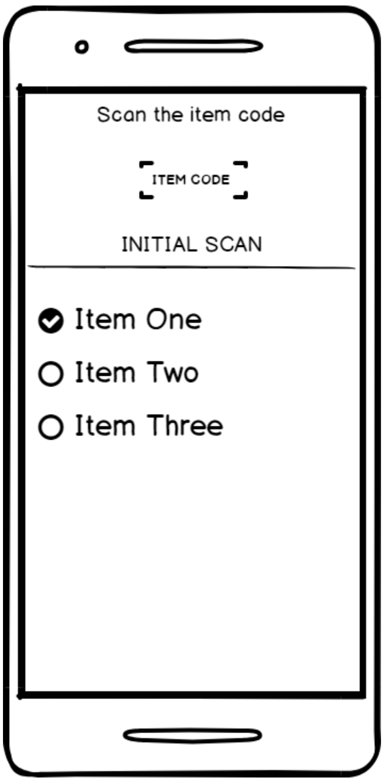

Figure B.2: Scanning device - mock 2

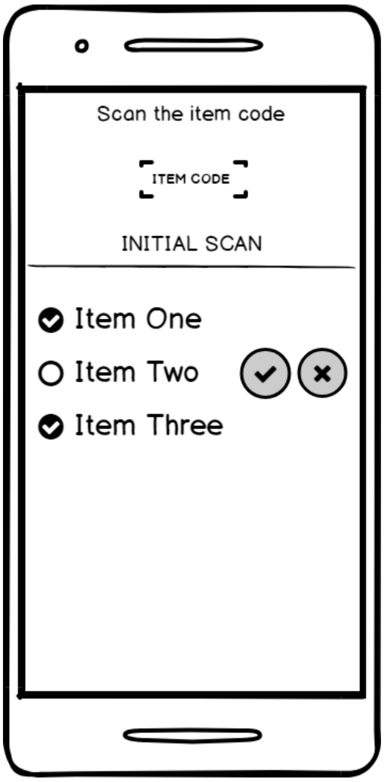

Figure B.3: Scanning device - mock 3

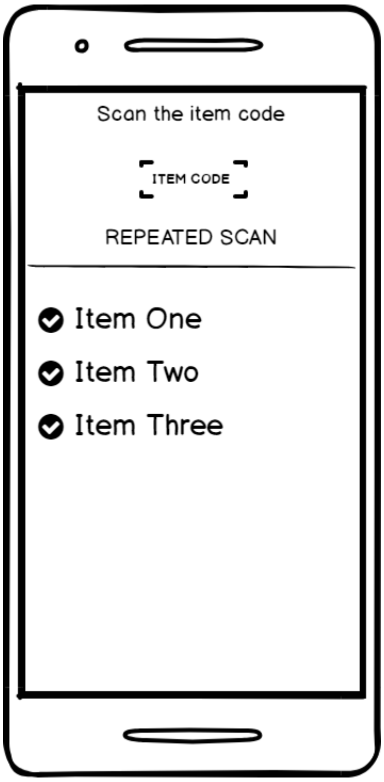

Figure B.4: Scanning device - mock 4

Appendix C

User interface of the scanning device

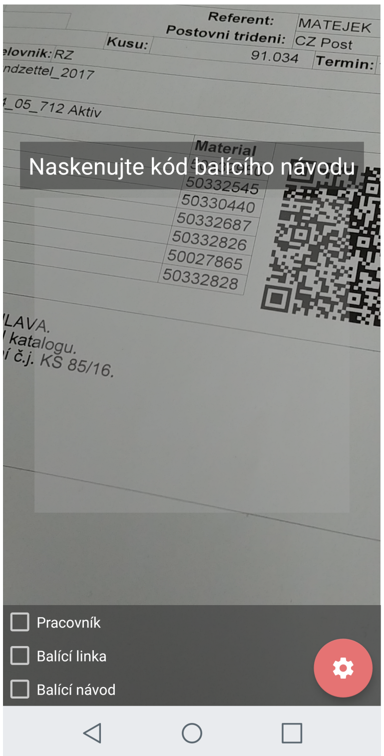

Figure C.1: Scanning device - screen 1

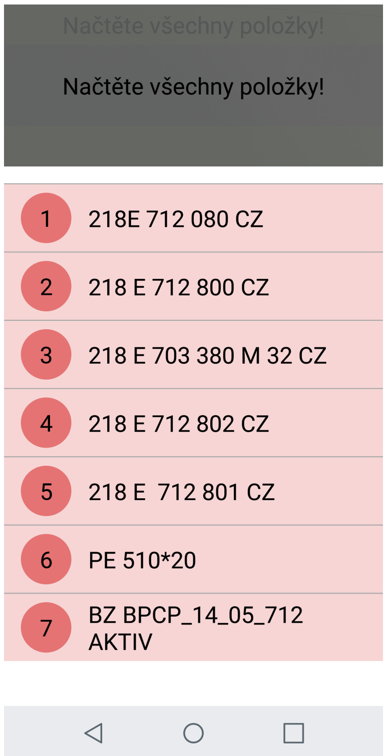

Figure C.2: Scanning device - screen 2

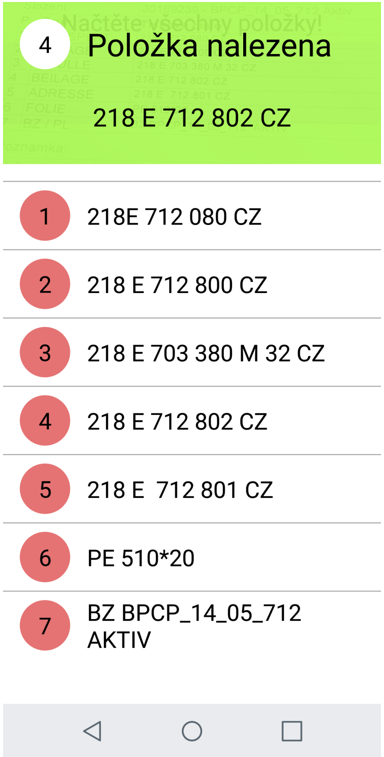

Figure C.3: Scanning device - screen 3

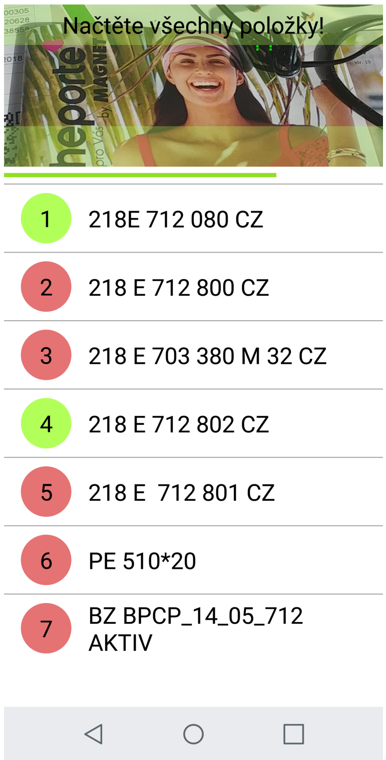

Figure C.4: Scanning device - screen 4

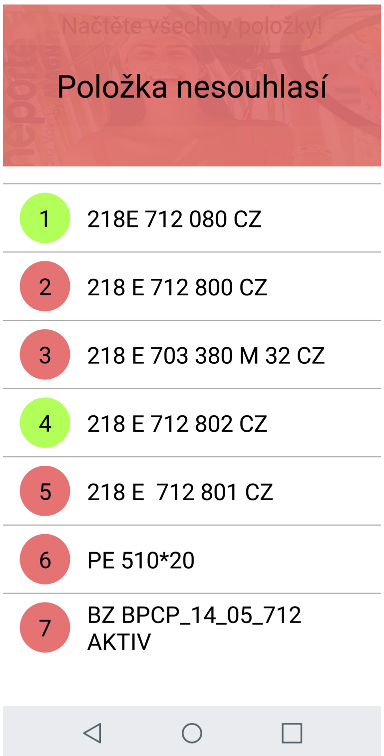

Figure C.5: Scanning device - screen 5

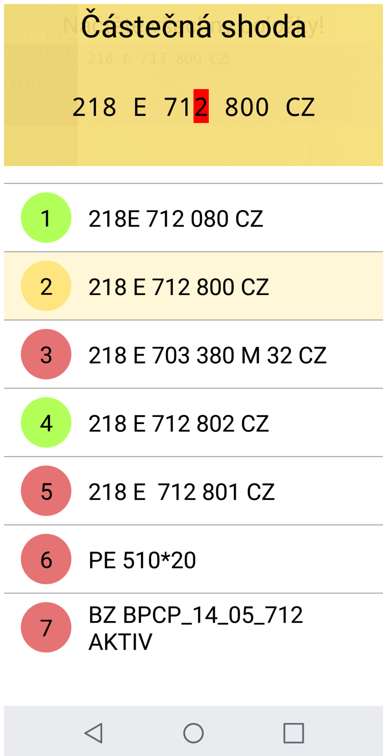

Figure C.6: Scanning device - screen 6

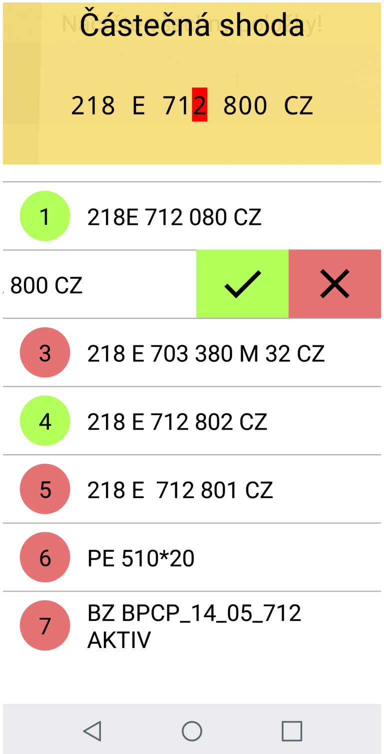

Figure C.7: Scanning device - screen 7

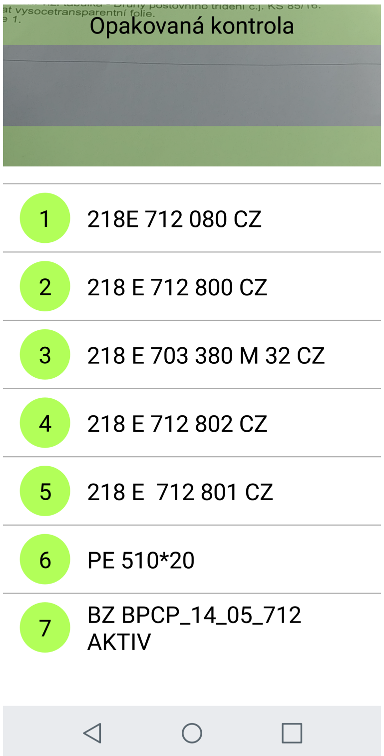

Figure C.8: Scanning device - screen 8

Appendix D

Usability testing logs

D.1 Participant 1

1. The verification process is beginning and the participant picks up the scanning device.
2. The participant struggles for a while with scanning the identifiers as he is not familiar with the QR codes.
3. Then he understood and points the camera to QR code identifier.
4. During the scanning he maintains the order specified in the bottom part of the screen, even it is not required.
5. He understood the system feedback when some of the identifiers is successfully scanned.
6. When all of the identifiers are scanned the scanning device shows the item verification screen.
7. He says that he is going to begin the verification by pointing the camera preview at the item identifier according to the instructions shown on the screen.
8. He struggles with aiming the camera, especially finding the correct distance for the camera to be focused, yet big enough for the scanning device.
9. Finally he finds out the camera capabilities of the scanning device.
10. The participant verifies the item types and prepares items into stackers. He maintains the item order as in the previous screen even it is not required.
11. Due to the incorrect aim of the camera a yellow alarm appears while scanning the item with correct code.
12. The participant examines the screen, but haven't noticed the characters with the red background at first sight.
13. He begins the manual verification of the item code finds out the difference.

14. He presses the confirmation button after he receives the items of correct type (verified manually).
15. At the item with the modified code, he proceeds similarly but realizes that the red background marks the differing characters.
16. Due to the previously invoked yellow alarm he understood the method of invoking the manual confirmation option by swiping left but admits he would not know it by himself.
17. The initial verification is completed, and the repeated verification begins.
18. User is familiar with all control elements, and therefore the verification runs seamlessly
19. All items are repeatedly verified, and the test is ended.

D.2 Participant 2

1. The participant picks up the scanning device and begins the process.
2. The participant points the camera of the scanning device to the packaging machine identifier.
3. She understood the system feedback and continued with scanning remaining identifiers.
4. After she loads all of the identifiers are, she is transferred to the verification screen.
5. She explores the screen for a while and then starts pointing the camera at the item identifier.
6. A little time is needed to find the correct distance and angle for the camera, but after a while, the verification went seamlessly.
7. The participant is supposed to verify item without item identifier. She tries standard gestures used in the operating system such as long press etc.
8. The third gesture she tried was the correct one - the swipe left. She said this gesture is familiar from the email client.
9. She continues with verification without any problems.
10. A yellow alarm occurred during verification.
11. She examined the screen for a second but then quickly understood the meaning of the characters with the red background and performed the manual verification by comparing the characters manually.
12. She discovers the difference and marks that scanned item type does not belong into the package.

13. After she requested and received the items of a correct type she scans the identifier and green alarm appears.
14. She has finished the initial verification and was brought to the repeated verification screen.
15. The verification went without any complications as she is now familiar with the scanning device.
16. All items were verified in a repeated verification, and the test ends.

D.3 Participant 3

1. The participant picks up the scanning device and the begins the verification.
2. He says he is not confident with the scanning device.
3. He reads aloud the instructions on the screen of the scanning device.
4. The participant tries to scan the packaging instructions by scanning the whole paper with packaging instructions.
5. The recognition of the identifier takes a while as the camera is not directly aimed to the QR code.
6. He was acquainted with the QR codes and their appearance after this incident.
7. The scanning of other identifiers is without any complications.
8. The participant does not follow the order in the list of identifiers to scan at the bottom of the screen.
9. Item verification screen is shown, and the participant reads aloud the instructions on the top part of the screen.
10. He tries to scan the item identifier according to the instructions.
11. The participant complains about the complexity of the camera aiming while pointing the camera to the identifier of the first item.
12. After several seconds the application shows the green alarm and the first item is verified.
13. He continues with verification of next items.
14. Due to his shaky hands the scanning takes a little longer.
15. The participant says that the bounding box of the text block helps him with camera aiming.
16. The participant picks up the item without the identifier and is supposed to perform the manual verification.

17. After a while of screen exploration, he discovers the given item in the list.
18. He tries to find out how to mark the item as checked, by randomly tapping the screen.
19. The participant got stuck on this step and was not able to continue in the process as he is not familiar with swipe gesture needed for unveiling the manual verification option.
20. Since he was told how to unveil the manual verification option the verification continued well.
21. He said that he would not be able to find out the swipe gesture by himself, but the color scheme and icons used at manual verification option are understandable.
22. He proceeds with verification of remaining items.
23. The participant picks up the item with the corrupted identifier and tries to verify it.
24. He is confronted with the yellow alarm and tries to figure the meaning of the alarm.
25. He compares the item identifier with identifier shown on the screen. Then he realizes that the red background marks the characters that are misplaced.
26. The participant correctly verifies the item does not belong into package and requests the items of the correct type.
27. He proceeds with verification of remaining items.
28. The initial check is completed, and the participant says that he feels much confident with the scanning device.
29. The repeated verification is started.
30. The participant verifies the items, but the way he aims the camera is still unconfident.
31. All items are repeatedly verified.
32. The test ends.

Appendix E

Server API description

```
swagger: '2.0'
info:
  description: Logging API
  version: "1.0.1"
  title: Logging API
  contact:
    email: me@tomaspikous.cz

  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html

tags:
- name: developers
  description: Operations available to regular developers
  paths:
  /user:
    post:
      tags:
      - developers
      summary: Creates an user in the system
      operationId: createUser
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: User
        required: true
        description: "User entity to be created"
        schema:
          $ref: '#/definitions/User'
      responses:
        201:
          description: User created
          schema:
            $ref: '#/definitions/User'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /user/{UUID}:
    get:
      tags:
      - developers
      summary: Gets user from system by UUID
      operationId: getUser
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the user"
      responses:
        200:
          description: User returned
          schema:
            $ref: '#/definitions/User'
        400:
          description: Bad request
          schema:
```

```
          $ref: '#/definitions/Message'

    put:
      tags:
      - developers
      summary: Updates user in system by UUID
      operationId: updateUser
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the user"
      - in: body
        name: User
        required: true
        description: "Updated user entity"
        schema:
          $ref: '#/definitions/User'
      responses:
        200:
          description: User updated
          schema:
            $ref: '#/definitions/User'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

    delete:
      tags:
      - developers
      summary: Deletes user from system by UUID
      operationId: deleteUser
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
        - in: path
          name: UUID
          type: string
          required: true
          description: "UUID of the user"
      responses:
        200:
          description: User deleted
          schema:
            $ref: '#/definitions/Message'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /packaging-instruction:
    post:
      tags:
      - developers
      summary: Creates a packaging instructions in the system
      operationId: createPackagingInstructions
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: PackagingInstructions
        required: true
        description: "Packaging Instructions entity to be created"
        schema:
          $ref: '#/definitions/PackagingInstructions'
      responses:
        201:
          description: Packaging instructions created
          schema:
            $ref: '#/definitions/PackagingInstructions'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /packaging-instruction/{UUID}:
    get:
      tags:
      - developers
      summary: Gets packaging instructions from system by UUID
      operationId: getPackagingInstructions
      consumes:
      - application/json
      produces:
```

```yaml
    - application/json
    parameters:
    - in: path
      name: UUID
      type: string
      required: true
      description: "UUID of the packaging instructions"
    responses:
      200:
        description: Packaging instructions returned
        schema:
          $ref: '#/definitions/PackagingInstructions'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

  put:
    tags:
    - developers
    summary: Updates packaging instructions in system by UUID
    operationId: updatePackagingInstructions
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: path
      name: UUID
      type: string
      required: true
      description: "UUID of the Packaging Instructions"
    - in: body
      name: Packaging instructions
      required: true
      description: "Updated packaging instructions entity"
      schema:
        $ref: '#/definitions/PackagingInstructions'
    responses:
      200:
        description: Packaging instructions updated
        schema:
          $ref: '#/definitions/PackagingInstructions'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

  delete:
    tags:
    - developers
    summary: Deletes packaging instructions from system by UUID
    operationId: deletePackagingInstructions
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the Packaging instructions"
    responses:
      200:
        description: Packaging instructions deleted
        schema:
          $ref: '#/definitions/Message'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'


/item-type:
  post:
    tags:
    - developers
    summary: Creates an item type in the system
    operationId: createItemType
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: body
      name: ItemType
      required: true
      description: "Item Type entity to be created"
      schema:
        $ref: '#/definitions/ItemType'
    responses:
      201:
        description: Item Type created
```

```
        schema:
          $ref: '#/definitions/ItemType'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

/item-type/{UUID}:
  get:
    tags:
    - developers
    summary: Gets Item type from system by UUID
    operationId: getItemType
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: path
      name: UUID
      type: string
      required: true
      description: "UUID of the Item type"
    responses:
      200:
        description: Item type returned
        schema:
          $ref: '#/definitions/ItemType'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

  put:
    tags:
    - developers
    summary: Updates Item type in system by UUID
    operationId: updateItemType
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: path
      name: UUID
      type: string
      required: true
      description: "UUID of the Item type"
    - in: body
      name: Item type
      required: true
      description: "Updated Item type entity"
      schema:
        $ref: '#/definitions/ItemType'
    responses:
      200:
        description: Item type updated
        schema:
          $ref: '#/definitions/ItemType'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

  delete:
    tags:
    - developers
    summary: Deletes Item type from system by UUID
    operationId: deleteItemType
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the Item type"
    responses:
      200:
        description: Item type deleted
        schema:
          $ref: '#/definitions/Message'
      400:
        description: Bad request
        schema:
          $ref: '#/definitions/Message'

/packaging-line:
  post:
    tags:
    - developers
    summary: Creates an Packaging line in the system
```

```
      operationId: createPackagingLine
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: PackagingLine
        required: true
        description: "Packaging line entity to be created"
        schema:
          $ref: '#/definitions/PackagingLine'
      responses:
        201:
          description: Packaging line created
          schema:
            $ref: '#/definitions/PackagingLine'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /packaging-line/{UUID}:
    get:
      tags:
      - developers
      summary: Gets Packaging line from system by UUID
      operationId: getPackagingLine
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the Packaging line"
      responses:
        200:
          description: Packaging line returned
          schema:
            $ref: '#/definitions/PackagingLine'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

    put:
      tags:
      - developers
      summary: Updates Packaging line in system by UUID
      operationId: updatePackagingLine
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: path
        name: UUID
        type: string
        required: true
        description: "UUID of the Packaging line"
      - in: body
        name: Packaging line
        required: true
        description: "Updated Packaging line entity"
        schema:
          $ref: '#/definitions/PackagingLine'
      responses:
        200:
          description: Packaging line updated
          schema:
            $ref: '#/definitions/PackagingLine'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

    delete:
      tags:
      - developers
      summary: Deletes Packaging line from system by UUID
      operationId: deletePackagingLine
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
        - in: path
          name: UUID
          type: string
          required: true
          description: "UUID of the Packaging line"
```

```yaml
      responses:
        200:
          description: Packaging line deleted
          schema:
            $ref: '#/definitions/Message'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /synchronize-log:
    post:
      tags:
      - developers
      summary: creates a verification log in the system
      operationId: synchronizeVerificationLog
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: VerificationLog
        description: Log entity to be stored
        schema:
          $ref: '#/definitions/VerificationLog'
      responses:
        201:
          description: Log stored
          schema:
            $ref: '#/definitions/Message'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

  /synchronize-logs:
    post:
      tags:
      - developers
      summary: creates multiple verification logs in the system
      operationId: synchronizeVerificationLogs
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: logs
        description: DTO containing multiple log entities to be stored
        schema:
          $ref: '#/definitions/VerificationLogs'
      responses:
        201:
          description: Logs stored
          schema:
            $ref: '#/definitions/Message'
        400:
          description: Bad request
          schema:
            $ref: '#/definitions/Message'

definitions:
  VerificationLog:
    type: object
    required:
    - user
    - packagingLine
    - packagingInstructions
    - manual
    - itemCode
    properties:
      user:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      packagingLine:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      packagingInstructions:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      manual:
        type: boolean
      status:
        type: string
        enum: [NEW, INITIAL_SCAN,REPEATED_SCAN]
      timestamp:
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
      itemCode:
```

```
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851

  VerificationLogs:
    type: object
    required:
    - logs
    properties:
      logs:
        type: array
        items:
          $ref: '#/definitions/VerificationLog'

  PackagingLine:
    type: object
    properties:
      uuid:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      description:
        type: string
        example: Packaging line 16 located in the factory in Brno

  User:
    type: object
    properties:
      uuid:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      firstName:
        type: string
        example: John
      lastName:
        type: string
        example: Doe
      role:
        type: string
        enum: [OPERATOR,MANAGEMENT,EXTERNAL_SYSTEM]

  PackagingInstructions:
    type: object
    properties:
      uuid:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      description:
        type: string
        example: Package for companyXY from 1.1.2018
      followItemOrder:
        type: boolean
        example: true
      items:
        type: array
        items:
          $ref: '#/definitions/ItemType'

  ItemType:
    type: object
    properties:
      uuid:
        type: string
        format: uuid
        example: d290f1ee-6c54-4b01-90e6-d701748f0851
      itemIdentifier:
        type: string
        example: MRC180XH/KV1
      order:
        type: number
        example: 0

  Message:
    type: object
    properties:
      message:
        type: string
        example: "Some API message"


host: virtserver.swaggerhub.com
basePath: /tomaspikous/loggingserver/1.0.0
schemes:
 - https
```

Appendix F

Content of the attached CD

Filename	Description
server/	server source files
client/	client source files
thesis/	LaTeX thesis source files
README.txt	Manual
MT-Tomas-Pikous-2018.pdf	Text of the Master's thesis

Table F.1: Content of the attached CD