Wearables in Logistics - Demo Facility

Final report of bachelor thesis

Submitted by Lukas Rolle

In fulfilment of the requirements for the degree
Bachelor of Science in Informatics
To be awarded by the
Fontys Hogeschool Techniek en Logistiek

Venlo, June 8, 2017

Information Page

Student Information

Name:	Lukas Rolle
Date of Birth:	23 March 1994
Place of Birth:	Moers, Germany
Student Number:	2310309
Study Course:	Informatics: Software Engineering

Thesis Information

Time frame:	01. February - 30. June 2017
Date of Delivery:	28 March 2017

Company Information

Name:	Fontys Hogeschool Techniek en Logistiek
Address:	Tegelseweg 255
Postal code:	5912 BG
City:	Venlo
Country:	Netherlands

Educational Institution

Name:	Fontys Hogeschool Techniek en Logistiek
Address:	Tegelseweg 255
Postal code:	5912 BG
City:	Venlo
Country:	Netherlands

Examination Committee

Company Supervisor:	Stefan Sobek
Supervising Lecturer:	Thijs Dorssers
Examinator:	Ferd van Odenhoven
External Representative:	J. Janssen

Statement of authenticity

I hereby solemnly declare for this submitted work, that:

- I myself wrote this internship report, without the assistance of any third party,
- I did not cut and paste any information (text, figures, diagrams, tables,..) from others without appropriate use of quotation marks and direct reference to their work,
- I did not re-word the ideas of others without proper and clear acknowledgement,
- I did not make use of ideas or suggestions that originated from others and claim these as my own,
- I did not include words from other's work without permission.

I am fully aware that any violation of the above will be declared fraud and may result in disadvantageous consequences for me (for example withdrawal of study credits and, in case of a repeated violation, withdrawal of complete study units). If fraud can be proved, I will be required to bear the costs of investigation into and sourcing of the original document.

Name: Lukas Rolle

Place / Date: Venlo, June 8, 2017

Signature:

Abstract

Many small- and medium sized enterprises have difficulties in trying out new technologies, as they often just do not have the needed resources to spend on the newest technology. The LOGwear project aims to give these companies the possibility to stay competitive in that area, it tries to take generalized logistics processes and combines these with wearables and make the results available to everyone.

This thesis is about the creation of a generalized reference model, that allows everyone a head start on creating their own application with a wearable in the area of logistics. Furthermore a demo facility is planned to be created to allow everyone that is interested in adopting the wearable technology to get hands-on experience. This demo facility is a physical area, where interested can come to visit and get a demonstration of how a process could look like when using a wearable for it. The environment for the demo facility is trying to mock a real world logistics company as closely as possible, using processes from logistics companies as a base.

This should allow companies, that are interested in adopting new technology, to inform themselves about these technologies easily, find technologies they think are interesting for their way of working, see how the technology actually works in a hands-on environment and then make an educated decision if the adoption of a wearable is something that could help improve their own processes.

Kleine- und Mittelständische Unternehmen haben oft Probleme damit neue Technologien auszuprobieren, da die dazu nötigen Ressourcen oft nicht vorhanden sind um die neuesten Technologien auszuprobieren. Das LOGwear Projekt versucht diesen Unternehmen die Möglichkeit zu geben in dieser Umgebung trotzdem wettbewerbsfähig zu bleiben. Es versucht standardisierte logistische Prozesse zu nehmen und Sie mit wearables zu verbinden und stellt die Ergebnisse frei zur Verfügung.

Diese Abschlussarbeit beschäftigt sich mit der Erstellung eines standardisierten Referenzmodells, das eine schnellere Erstellung einer eigenen Anwendung mit wearables erlaubt. Weiterhin ist eine Testeinrichtung geplant die kleinen- und mittelständischen Unternehmen die daran interessiert sind, die Möglichkeit gibt wearables selbst auszuprobieren. Die Testeinrichtung ist eine Umgebung zu der interessierte gehen können, um selbst sehen oder ausprobieren können ob ein bestimmtes wearable etwas für ihre Unternehmensstruktur ist. Es wird versucht mit der Testeinrichtung die tatsächliche Umgebung eines Logistikunternehmens nachzuahmen, in dem man wirkliche logistische Prozesse als Basis für die Testeinrichtung nimmt.

Damit sollte interesierten Unternehmen erlaubt werden, sich auf einfache Art und Weise über neue Technologien zu informieren, Technologien zu finden die in die Unternehmensprozesse passen könnten, zu sehen wie diese Technologien tatsächlich funktionieren und daraufhin eine informierte Entscheidung zu treffen, ob diese Technologie tatsächlich etwas ist, was Ihre Unternehmensprozesse verbessern könnte.

Contents

List of Figures

List of Tables

Glossary

battery pack a device used to store a bigger amount of power, generally able to charge a phone or similar devices multiple times.

cloud computing a technique that provides computing resources in a virtual server environment. Giving customers more or less computing resources depending on the demand of each specific user.

haptic feedback feedback given to a user via touch, most of the time being vibrations. Differences in haptic feedback can be the amount, place or intensity of the vibrations.

hot swap is the action of swapping a part of a device for a replacement, while the device is still powered on. For a phone it could be swapping the battery for another one, without turning the phone off, the phone has to support this feature.

package in software engineering terms, a package is a collection of classes that are in general grouped by the responsibilities they have.

parcel a package in the logistics sense, a physical package that is a part of an order.

reference architecture a template solution in a domain that gives multiple models, sets of functions and classes to describe in detail how something is supposed to work.

reference model an abstract model used to describe the general design of an application in a specific environment. A reference model is in general technology-agnostic and can be extended to ones wishes.

ring scanner a scanner that is worn on one or multiple fingers similar to a ring. This allows workers to lay the scanning device down when needing both hands for work.

sandbox an environment where something can be tested without having an influence on anything else. In software engineering, an environment where an application can be tested without damaging the live version.

smartglasses a set of glasses falling under the category of wearable technology, generally able to project something on the glasses to output information to the user.

smartwatch a watch falling under the category of wearable technology, generally able to communicate with a phone to have information directly available on the wrist.

wearable a piece of technology that can be worn on the body.

List of Abbreviations

API Application Programming Interface.

CASE Computer-Aided Software Engineering.

DB Database.

EU European Union.

FHTenL Fontys Hogeschool Techniek en Logistek.

HPU Holographic Processing Unit.

HSNR Hochschule Niederrhein.

ID Identifier.

IDE Integrated Development Environment.

MSSQL Microsoft Structured Query Language.

MWEIMH NRW Ministerium für Wirtschaft, Energie, Industrie, Mittelstand und Handwerk des Landes Nordrhein-Westfalen.

QR Quick Response.

REST Representational State Transfer.

RFID Radio-Frequency Identification.

SDK Software Development Kit.

SME Small and Medium-sized Enterprises.

UML Unified Modeling Language.

VCS Version Control System.

WLAN Wireless Local Area Network.

WMS Warehouse Management System.

WP Work Package.

1 Introduction

This thesis is written as the completion of the study course software engineering at the Fontys Hogeschool Techniek en Logistiek in Venlo, Netherlands. The graduation project is conducted at the LOGwear research project, the research project itself will be explained in section 2.1. The thesis is written over the course of five months and is documenting the thought and creation process of the project, of creating a demo facility for the usage of a wearable in a logistics process.

This report is written during the second month of the thesis, therefore a lot of the tasks that should be done are not finished and it can only be explained how they are planned.

As this thesis was written on a software engineering topic, pieces of code will occur throughout the thesis, single words, like variables, or classes will be written in a `mono-spaced-font`.

Furthermore the word package in this report is always meant as a software package and never as the package that is used in the logistics sector, the word parcel is used in this context.

Overview

The following chapters in this report will contain these topics:

Context and Scope
: In chapter 2 the context of the project will be elaborated, naming the involved parties and what the research project is about. Furthermore the scope of the thesis will be defined, including demarcation. The general information, including project management details are explained.

Research
: In chapter 3 the general research part that was done during the thesis will be explained. The results of the research will be named and the chosen Computer-Aided Software Engineering (CASE) tools listed.

Reference Architecture
: Chapter 4 contains the design process and the connected problems with the reference model.

Demo Facility
: Chapter 5 will contain the infrastructure, design and implementation of the demo facility that is to be created to showcase the possibilities of wearables in the area of logistics.

Conclusion
: Chapter 6 will contain the conclusion, reflection and a look into the future, showing how the project is planned to develop.

2 Context and Scope

This chapter contains the context of the project, including an explanation of the research project LOGwear in section 2.1 and the parties involved in it. Furthermore the general project management strategies will be explained in the following sections: section 2.2 explains the stakeholders of the thesis, section 2.3 goes over the risks, section 2.4 explains the quality management for the project and section 2.5 describes the definition of done. Finally, section 2.6 is about the time planning and scheduling done for the given project.

2.1 LOGwear

LOGwear is a research project that aims to bring wearables to the area of logistics, especially to Small and Medium-sized Enterprises (SME). It is a German-Dutch research project were multiple parties are cooperating to create results. Involved in this are two Universities of applied sciences, Fontys Hogeschool Techniek en Logistek (FHTenL) in Venlo as the lead partner, Netherlands and Hochschule Niederrhein (HSNR) in Krefeld, Germany.

Further on there are also multiple partner companies involved in the project, namely KLG Europe bv, Helmut Beyers GmbH and imat-uve GmbH. These partner companies are there to give the knowledge about logistics processes, as well as to verify and test the results.

The project is backed within the scope of the INTERREG Deutschland-Nederland initiative. It is backed by the European Union (EU), Ministerium für Wirtschaft, Energie, Industrie, Mittelstand und Handwerk des Landes Nordrhein-Westfalen (MWEIMH NRW) and the Provincie Limburg as well. The official kickoff meeting for logWear was in september 2016 and the project will run until march 2018.

The LOGwear project is consisting of three main Work Package (WP)s. (Logwear.eu, 2017)

Knowledge Base (WP1)
: The knowledge base is a platform that allows to exchange information between logistics companies which wearable can be used for which process. (Sander, 2017) (Canders, 2017)

Reference Architecture / Reference Model (WP2)
: The expected result for WP 2 used to be a reference architecture, this has internally changed to a reference model, the differences about these two and what is expected from the reference model can be found in chapter 4.

Demo Facility (WP3)
: The demo facility is the creation of a physical demo that allows SME to see the benefits of wearables for a demo process. The difference between the demo facility created during this thesis and WP 3 is, that the WP 3 is the implementation of a wearable solution at a pilot company, while the demo facility created during this thesis is in an enclosed environment that does not need to follow all constraints of deploying something at a company. Further details for the demo facility are described in chapter 5.

2.2 Stakeholder

Stakeholder Management involves identifying parties that are involved in the project. This ranges from people that are actively part of the development of the project or companies that might be interested in the end result. In the table 2.1 the most important stakeholders can be found. The stakeholders will themselves will be further explained in sections 2.2.1, which will explain the internal stakeholders, and 2.2.2 will explain the external stakeholders involved in the project.

Nr	**Stakeholder**	**Company / Institution**	**Internal / External**	**Level of Interest**	**Level of Influ-ence**	**Potential man-agement strate-gies**
1	Employer (Em)	FHTenL	Internal	Medium	High	Keep Satisfied
2	Student Workers (SW)	FHTenL	Internal	Medium	Low	Keep Informed
3	Graduation Student (GS)	FHTenL	Internal	High	High	Key Player
4	Project Manager (PM)	FHTenL	Internal	High	High	Key Player
5	Company Supervisor (CS)	FHTenL	Internal	High	High	Key Player
6	Project Team (PT)	FHTenL	Internal	High	High	Key Player
7	Partner University (PU)	HSNR	External	Medium	Low	Keep Informed
8	Pilot Company (PC)	KLG	External	High	High	Key Player
9	Examiner (Ex)	FHTenL	External	Low	High	Keep Satisfied
10	Supervising Lecturer (SL)	FHTenL	External	Medium	Low	Keep Informed

Table 2.1: Stakeholder Register

Figure 2.1 can be used to visualize the importance of the stakeholders. The color is used to emphasize the importance that this stakeholder is properly managed.

2.2.1 Internal Stakeholders

Internal Stakeholders are parties that are a part of the team that is working directly on the project in one way or the other. In this section, the internal stakeholders mentioned in table 2.1 will be listed again and explained.

Employer

The employer in this project is an institution and not a single person. This does not change the fact, that the employer is interested in the project, as he is financing the project. Also the employer could change the outcome, if he is not accepting the proposed plans. It is important, that this party is kept satisfied as more work could be created when the plan has to change.

Student Workers

Student Workers are employed to help in the LOGwear project in general. While they currently are not involved with the process of the creation of the demo facility that might change in the future, therefore they should be kept informed.

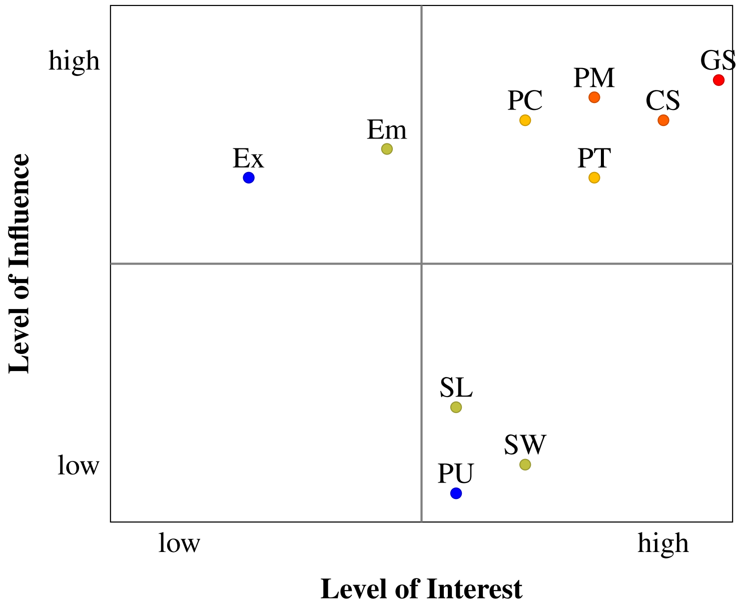

Figure 2.1: Stakeholder Graph

Graduation Student

The graduation student is the person mainly responsible for the development of the demo facility and therefore has a lot of responsibility and interest towards the project.

Project Manager

The project manager is responsible for the general planning of the project. Planning meetings with the different parties and coordinating them.

Company Supervisor

The company supervisor is looking over the progress of the graduation student and is giving advice if needed.

Project Team

The project team are the members of the team actively developing the application prototype and are involved in building the demo facility afterwards.

2.2.2 External Stakeholders

External Stakeholders are parties that are involved in the project, but are not a part of the team actively developing the project. In this section, the external stakeholders mentioned in table 2.1 will be listed again and explained.

Partner University

The partner university is also working on the LOGwear project, but on a different aspect. They might be interested in project of creating a demo facility, but probably will not interfere with it.

Pilot Company

The pilot company involved is the logistics company KLG. They bring in the highest amount of domain knowledge and are interested in the project to improve their own processes. They could influence the project easily by not approving the planned demo facility due to problems with how the logistics process is modelled.

Examiner

While the examiner is not involved in the project itself, the examiner will finally assesses the performance of the graduation student.

Supervising Lecturer

The supervising lecturer is there to answer questions and support the student from a software engineering standpoint. While not having a lot of influence on the project itself, the supervising lecturer is interested in what the student is doing and especially how he is doing it.

2.3 Risks

Risk management is about identifying risks and finding solutions to problems before they can occur. The list of risks can be found in table 2.2. The identified risks will increase as the project moves forward. Especially when a decision is made for the wearable and the process.

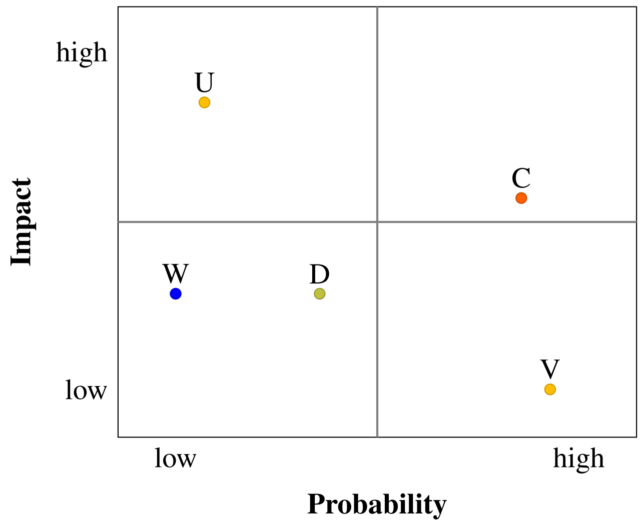

Figure 2.2: Risk Graph

In figure 2.2 the risks can be seen in a graph that shows their Probability and Impact again. The color emphasizes the amount of attention a risk should get, in order for the project to continue smoothly. The figure also shows more fine grained the probability and impact of the risks than just low, medium and high.

Nr	Risk Name	Description	Prob- ability	Impact	Root Cause	Potential Responses	Risk Owner
1	Wearable unavailable (W)	The wearable desired to be used in the demo facility is unavailable.	Low	Medium	The desired product is a prototype or similar.	Choosing a different wearable that is already readily available.	Graduation Student
2	Demo Area (D)	A demo area is in mind that could potentially be rented, but that could not be possible.	Medium	Medium	The owner of the place does not rent the area.	Researching possible places where the demo facility could be created.	Graduation Student
3	Unusable wearable (U)	A wearable is chosen that does not have the capabilities to fulfill the things that were planned with the demo facility.	Low	High	Too little research done on the wearables, or the researched material was wrong.	Altering the demo scenario to accommodate the problems with the wearable.	Graduation Student
4	Vocabulary unclear (V)	The vocabulary used in the logistics branch, especially abbreviations and acronyms might cause problems in communication.	High	Low	The graduation student has too little knowledge of the logistics branch, at the beginning of the project.	Asking questions if a word's or sentence's meaning is not clear.	Graduation Student
5	Communi- cation Problems (C)	When explaining a task, a phrase or word is understood differently from different parties.	High	Medium	The proper definition is not known to everyone and are expecting a different meaning from a given phrase or word.	When the misun- derstanding is discovered the different parties talk out what is expected from that term and come to a common understanding.	Project Team

Table 2.2: Risk Register

2.4 Quality Management

Metrics used to determine the quality of the project:

Code Coverage
: Code coverage is a useful metric showing the amount of code covered by unit tests. While they should not be the only way of testing an application of this size, they are still useful to see if a single components works on their own. Due to time constraints code coverage of 100% is most likely not achievable and a coverage of about 90% is aimed towards.

Language Conventions
: Programming languages usually have conventions on how the semantics of the code should look like to be considered code. These conventions will most likely be followed, and if there are any changes to that they will be listed here.

Documentation
: Documentation of a method will include the parameters involved, the return result, usage and general how something works. Furthermore documentation explaining the class and package will be created in a similar fashion. Furthermore created diagrams will be made available and also documented.

Performance Testing
: The wearable part of the application might need to be performance tested, as wearables tend to be less powerful than most computing devices normally used. This will be decided when the application is creating problems regarding to performance.

2.5 Definition of Done

A part of the software project is done, when it is fully designed, implemented, documented and tested. When that part of the application is passing all of these criteria it is added to a repository where the result is build. When that build is successful that part of the application is done, for the moment. When that part needs to be changed in the future the same procedure will be used again.

2.6 Planning

The initial schedule for the project can be seen in table 2.3. It is to be mentioned, that the schedule is subject to change as the project goes on. The project will be executed in a scrum-like way that is adapted to the group, given the group size of two developers.

Sprints	Logistics Processes & Wearables		Reference Architecture				Research Demo Facility		Demo Facility Design and Implementation									Creation Demo Facility			Buf-fer
Date	06.02	13.02	20.02	27.02	06.03	13.03	20.03	27.03	03.04	10.04	17.04	24.04	01.05	08.05	15.05	22.05	29.05	05.06	12.06	19.06	26.06
Week	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Table 2.3: Schedule

The schedule is divided in work packages that are to be executed, it is to be noted, that each work package could be split into multiple sprints in the future. In the following subsections the work packages will be explained.

make graphic readable, add more milestones

2.6.1 Logistics Processes & Wearables

This work package includes research about the given processes and wearables in general, as well as already choosing potential wearables that could be used to improve the process. The end result for this should be a decision on process and wearable. But the result for this could potentially take longer than this task is scheduled. The wearables should be ranked after getting hands-on experience on them, therefore some of them have to be ordered first.

2.6.2 Reference Architecture

A reference architecture should be created for a sample wearable application. What this work package contains is, the creation of diagrams which show the communication from a wearable to the Warehouse Management System (WMS) or something similar. What should not be created is a full reference architecture for a process that is implemented with a concrete wearable.

It is about creating the always needed layers when using a wearable in a way that supports most wearable solutions.

2.6.3 Research Demo Facility

This task includes researching what physical objects and what systems would be needed to create a demo facility that could showcase a single process with a single wearable. This also includes the gathering of knowledge of where the demo facility should be created and where to get the needed objects.

2.6.4 Demo Facility Design and Implementation

The work package includes the creation of the software design and implementation for the wearable and all aspects that are needed to fully showcase a process.

2.6.5 Creation Demo Facility

This task includes the physical creation of the demo facility. This means setting up shelves with packages to scan and put on a hand pallet truck. Setting up barcodes on the packages to scan. Setting up an environment that can showcase what is happening better to an audience.

3 Research

This chapter includes the general research that has been done for the project, which includes the research about some given processes and wearables that could be appropriate for the processes. The CASE Tools chosen at the current state of the project will also be discussed and what they are used for.

3.1 Processes

There were process already modelled and made available in the LOGwear project. They were created by working students at the FHTenL and revised through customer meetings. The naming of the processes was also done by the working students. The processes existing at the time of this report were:

- Order Picking High Rack
- Order Picking W3-5
- Goods Receipt and Put away

The decision for a single process was quickly made in cooperation with the company supervisor. The processes will just be briefly explained here, as it is not as important to fully understand the processes that were not chosen.

Order Picking High Rack has been discarded due to the nature of what should be created. A simulation that should model a real environment. Modelling a High Rack and usage of that would be too hard and would take too much time. add short description of what the process is about not just why it was discarded

Goods Receipt and Put away was another process to be considered, but was discarded. The modelling of it would be problematic due to the time delays in the tasks. Furthermore the process showed less potential for improvements with wearables. add short description of what the process is about not just why it was discarded

Order Picking W3-5 was the process chosen to improve with wearables as multiple possibilities to improve the process were discovered. Also the processes seemed rather easy to model with a demo facility by setting up one or two racks and placing multiple packages on there. This process was selected together with the company supervisor. This process can be seen in the appendix, in figure A.1 but the demo scenario will also be further explained in section 5.2.

3.2 Wearables

Researching wearables was a more problematic task, as the amount of wearables that could be used potentially is a lot higher. Some criteria were set in place for a wearable to be considered in the first place. A wearable has to be either already available, or freely available to order in europe from a trusted source. This leads to well known wearables like the Google Glass to be not even considered as they are not publicly available.

3.2.1 Criteria

Further criteria are more specific towards the chosen process. The criteria were divided into requirements and quantifiable criteria. The requirements a wearable, or a combination of wearables, must be able to fulfil are:

Scan ID
The ability to scan an Identifier (ID), could be a barcode, Radio-Frequency Identification (RFID) code, Quick Response (QR) code or something different.

Informing User
The ability to give information to the user, this can be done by audio, visual or haptic feedback.

Send Confirmations
The ability to send confirmations to the WMS, that part of the process has been completed.

Hands-free
The ability to operate the wearable without the need to take a device and put it back all the time. Operations should not need user hand input while packages are being handled or the hands are otherwise occupied.

On top of these criteria that the quantifiable requirements were:

Performance
The performance of a device is important as some wearables considered are not necessarily intended for the tasks in a warehouse and therefore some tasks like scanning different kinds of barcodes might be more problematic and need a lot of time if the device is not performant.

Cost
The cost is a factor to consider when the company possibly needs to order a large amount of these wearables.

Battery Life
A wearable should be able to sustain a whole day of working without the need to exchange batteries or swap wearables during a break. Worst case scenario would be to have the need to change the wearable device multiple times each day due to an empty battery.

Durability
The warehouse is not an environment where devices can be used that could break if a parcel graces or hits it.

Most of the quantifiable criteria were hard to actually quantify without having hands-on-experience with these devices. The reason behind this is, that most of the devices are either just released to the public or niche products leading to a small amount of information available besides information published by the manufacturers of the devices, which is generally not a good source of unbiased information.

3.2.2 Devices

Smartglasses seem to be the most promising type of wearable for the task, due to the possibilities it does give its user. The possibility for indoor navigation, scanning, exactly displaying the item location, constant display of information and further possible features that could be implemented using an always-on camera that is implemented in most devices. (Schwerdtfeger, 2009)

The smartglasses most interesting for this topic are:

- Epson-Moverio BT-300
- Microsoft HoloLens
- Vuzix M100

They are the newest publicly available glasses from some of the biggest manufacturers of smartglasses, that are currently publicly available.

Another solution including wearables for the order picking process is the combination of a ring scanner and a wrist-mounted computer. Wearables for this type of combination are:

- Zebra WT6000 + RS6000
- Honeywell Dolphin 75e + 8620 Wearable Ring Scanner

This combination could also be using a headset to add further functionality.

3.2.3 Decision

The wearables that were tested hands-on have been:

- Microsoft HoloLens
- Vuzix M100
- Honeywell Dolhin 75e + 8620 Wearable Ring Scanner

The other named wearables were also intended to be tested, but were not able to be delivered in a time that was considered acceptable to continue the project. Therefore the final weighted decision will only take these three wearables in consideration. The combination of Honeywell Dolphin 75e and the 8620 Wearable Ring Scanner will be counted as a single wearable for this comparison as they are intended to be used together. The general weighted decision for these wearables can be found in table 3.1.

		HoloLens		Vuzix M100		Honeywell	
Criteria	Weight	Value	Weighted	Value	Weighted	Value	Weighted
Performance	4	6	24	2	8	8	32
Cost	2	2	4	5	10	3	6
Battery Life	2	4	8	5	10	9	18
Durability	2	3	6	3	6	8	16
			42		34		72

Table 3.1: Weighted Decision Wearables

The wearables were given a value between one and ten depending on how well they performed in the hands-on test and the information available beforehand.

From the weighted decision we can see that the Honeywell combination comes out on top, which is not as surprising, as it is a device that was specifically designed for this and similar purposes. The performance of the Honeywell combination is rated as highly, because the ring scanner is handling the scanning, which can be a more expensive task for ordinary computing devices. The HoloLens is in front of the Vuzix M100 mainly because of the integrated Holographic Processing Unit (HPU) that is handling the mapping of the environment and most of the data that is incoming from the different cameras, therefore allowing the other components of the HoloLens to do different tasks with higher priority. Another factor here is that the HoloLens in general is using newer components that have a higher performance by themselves. The cost or price of the device is just the price of the devices.

The battery life between the HoloLens and the M100 is relatively similar, but the Vuzix solution has an included battery pack to increase usage time, which is also something that could be done with the HoloLens, but since charging and using a device at the same time is not the nicest solution. Therefore the score is pretty low for both. When using a battery pack, both devices might get through a workday, but without both need to be recharged multiple times a day. The Honeywell combination is able to sustain through a day of work, and if needed, the battery here can be hot swapped. The durability of both smartglasses is similar in points, as both were not necessarily designed in the warehouse. While the HoloLens might break a bit faster than the Vuzix M100, at least from how rugged both look and feel, the M100 just needs a small push to no longer be in the right position to view the monitor. The Honeywell combination is by far the most rugged and also can not disposition as easily, the biggest problem might be the connection between the ring scanner and the wrist-mounted computer. While not likely that could tear or be plugged out.

But that might not be the only things that could lead to a decision, an aspect that is not explored here is possibilities. But the possibilities that a wearable has, are not that easily measured and therefore are ignored for the wearable comparison at this point in time. Therefore the Honeywell Dolphin 75e + 8620 Wearable Ring Scanner has been chosen as a wearable to implement the demo scenario explained in section 5.2.

3.3 CASE Tools

The CASE tools chosen here are only for the parts already decided on. As a chosen wearable could introduce a lot more CASE tools to the project. The already chosen CASE tools can be seen in table 3.2.

Program	**Usage**
UMLet	UMLet is a free, open-source Unified Modeling Language (UML) tool. It is used to create UML diagrams and is extensible by user created, custom elements. (Umlet.com, 2017) It was used to create the diagrams that were created during the thesis.
JIRA	JIRA is a tool developed by Atlassian. It is an issue tracking and project management tool. During the thesis it is used as a tool to plan sprints and track bugs and issues, as well as having a backlog of features that should still be implemented.
Bitbucket	Bitbucket is a tool developed by Atlassian. It is a Version Control System (VCS) solution that uses git as an underlying system and is aiming to give companies a secure and private git repository to use for proprietary projects. During the thesis Bitbucket is used as a VCS for the different parts of the project.
Visual Studio	Integrated Development Environment (IDE) developed by Microsoft for programming languages managed by them. For this project used to implement the database connection and the Representational State Transfer (REST) Application Programming Interface (API).
Mobility SDK for Android	A Software Development Kit (SDK) used to develop for the Honeywell Dolphin 75e. This is provided by Honeywell to allow usage of the unique components in their devices.
Azure Cloud	A cloud computing environment by Microsoft. In this project used to deploy the database for the mock WMS, the database connector and the REST interface to connect to it.

Table 3.2: CASE tools

4 Reference Model

The reference model of the LOGwear project is a model that shows how the communication from a wearable to the WMS or some other system might work. This does not mean that every communication necessarily needs to look like displayed in the reference model, but that in general every wearable is able to communicate with an underlying system in the way it is displayed. The model can be seen in figure 4.1.

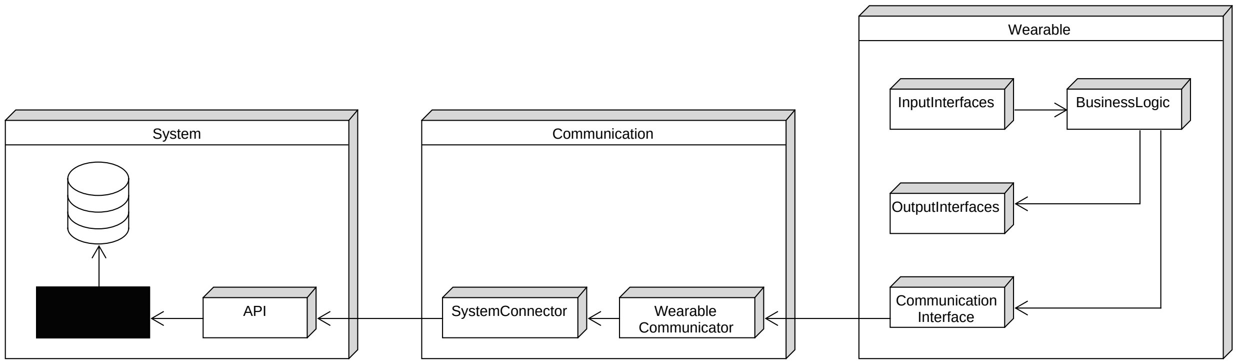

Figure 4.1: Reference Model LOGwear

4.1 Definition

A reference model is an abstract design used to help others understand the general concept and the relationships between existing entities of a specified environment. Furthermore a reference model, in general, does not model anything with a specific technology in mind and rather models everything as general as possible. This is done so that when creating an architecture around the reference model. The reference model can be used as a template to start working with. Not as a constraint, that holds the architects back. When implementing the reference model with a specific technology, it is needed to change existing parts or add new parts to fit the given constraints. A reference model as is, is not directly implementable due to the abstract nature.

The aim for a reference model is to standardize the way how developers in the future implements an application in the given domain, regardless of the used technologies.(Oasis-open.org, 2017)

4.2 Requirements

The reference model is an abstract construct, therefore the functional requirements towards it are taken as example to prove the validity of the model. The functional requirements are listed in the form of use cases The non-functional requirements are towards the model itself and not towards an implementation of it.

4.2.1 Functional Requirements

The functional requirements represents example cases that should be possible to implement with the reference model. The cases that will be given do not need to be fulfilled all at once, but all of them should be possible with the reference model. Given in the following list are short descriptions of the use cases, while the full use cases themselves can be seen in the appendix, section A.2. The example cases given here are based on the demo scenario chosen, but more use cases can be defined for different scenarios.

Get Order
An order document is fetched and in some way displayed to the worker.

Order Confirmation
The order the worker is currently working on is being worked on and the parts of the order is being confirmed. When all parts of the order are finished the whole order can be confirmed.

Order Control
When an order is being picked, the wearable is supporting the worker in counting the right number of parcels and picking the correct article in the first place.

4.2.2 Non-Functional Requirements

The non-functional requirements towards the reference model are the most important details that were taken in consideration when the reference model was designed. They can be seen in the following list:

Documentation
The reference model needs to be properly documented. The advantages and disadvantages of the design need to be properly explained to a person that potentially wants to implement a system based on it.

Extensibility
The reference model needs to be extensible since some companies might have some more requirements towards their system than is intended for a general case. Adding functionality to the reference model should be possible.

Modifiability
The existing ideas of the reference model should be modifiable. Companies are going to use different wearables and infrastructure, and the reference model should be adjustable to fit a lot of possibilities without the need to create a completely new system architecture.

4.3 Design

As can be seen in figure 4.1 the reference model is divided into three different packages that all fulfil different responsibilities. First it will be described what this reference model can help to create. Therefore the general thought behind the whole model will be explained and afterwards the three packages `system`, `communication` and `wearable` will be explained on their own.

The concept is simple, a wearable is connected to a communication layer, that then again connects to a system, that could be anything, as long as it has an API. The arrows are not indicating an information flow but rather an instruction flow. The box that the arrow leaves does invoke an instruction in the box that the arrow points to. The sources of information are generally the `InputInterfaces`, and the incoming information is spread from there. Depending on the incoming information, actions are invoked.

The reference model is purposely minimalistically designed, to allow most infrastructures and wearables, to apply the model, with as few changes as possible. If for example a new WMS was used the only component that needs to be changed is the `SystemConnector` in the `communication` package. In that case, the wearable that are in use, can also continue to work, just like they normally would, without the need for a new version to be deployed on all of them.

4.3.1 Wearable

The `wearable` package is representing the actual physical wearable, or a set of wearables, that a worker is using. This could be either smartglasses, smartwatches, a ring-scanner or some other wearable. It could also be a combination of wearables that is used in order to fulfil a certain task. In such cases it is still possible to stick to the reference model either by changing the existing model, with multiple `BusinessLogic` classes in the different wearable and a manager that could handle that. Or it could be, that even when using multiple wearables at once, a single wearable is handling the business logic of all wearables. Then the other wearables could be addressed by just their input- and output- interfaces.

Further on the wearable package is again held as simple as possible. The `InputInterfaces` are there to get information from the outside world and the `OutputInterfaces` are there to somehow represent information to the outside world. The `BusinessLogic` is there to process the incoming data and invoke the appropriate actions, that could be either displaying some information to the user or making a call to the underlying system through the `CommunicationInterface`.

The `CommuncationInterface` is the means of communication with an outside computer source, this could be radio, bluetooth, REST via Wireless Local Area Network (WLAN) or some other way of communication. The `WearableCommunicator` in the `Communication` package just needs to be able to receive and understand the messages.

4.3.2 Communication

The `communication` package is existing due to the different technologies used from wearables to connect to other computing devices. The communication layer might also be the only place, where information can be fully controlled by the developer, this is especially interesting when the `system` is controlled by a third-party. Also needed actions can be taken, if the incoming data has to be transformed into a different format, before either the `API` or the `wearable` can understand it.

The `communication` package could potentially be removed, if the wearable has the needed technology in place to directly connect to the API of the given system. But this is in general not recommended, as the communication layer allows the developer to create a more standardized flow of information.

4.3.3 System

The `system` package here can be something like a WMS of the company. A system that is mostly a black box with an API and a database. Most of the time, the `system` cannot be changed by the developer, therefore the developer needs to use the possible ways to connect to the given API.

4.4 Variations

add infromation on how the model could be changed to implement push messages.

Differences in how the business logic could be handled an what it could do, web app, effectively removing business logic in the wearable

4.5 Problems

The main problem that occurred during the creation of the reference model was a problem in communication, regarding the names of reference model and reference architecture. The initial task was understood to create a general-purpose reference architecture, that would connect a wearable to some kind of system. During the process of creating the different diagrams for the reference architecture and trying to validate them using code. The problem became obvious that trying to go deeper than the now given reference model, as seen in figure 4.1, was impossible to do for a general-purpose implementation.

This is the case because of the nature of the given problem. Being able to use any wearable with any system, for any task. Given that two wearables might have completely different sets of input- and outputinterfaces those could not be defined. The communication interface could be different, which leads to being unable to define a communication standard, while the possibility of any task gives no single action that will always be the same.

5 Demo Facility

Physical creation of the demo facility that showcases the possibilities of the chosen wearable in the process of order picking in a warehouse. As the demo facility is not yet existing the implementation cannot be discussed and explained here, this chapter will therefore focus on the planned infrastructure, the existing design and what is planned for the demo facility in the future.

There is also a major difference between the demo facility task that will be explained in this report and the task given in the logwear website. (Logwear.eu, 2017) The task that will be executed here will be creating a demo facility in a physical sandbox environment and the task explained in the work package on the logwear homepage is about implementing the improved process, using a wearable, at a pilot company and observing the results.

5.1 Infrastructure

The infrastructure of the demo facility is divided into multiple areas:

Mock WMS
: A mock WMS that allows to store different orders and edit them to have a more realistic demo. While not all functionality has to given, the data has to be persistent and easily resettable to allow showcasing a demo case multiple times. The mock WMS is living on an azure server and is a Microsoft Structured Query Language (MSSQL) database.

Rest API
: A REST API that is used to connect to the mock WMS from an outside perspective, in this case from the communication layer, see subsection 4.3.2.

Communication Layer
: The communication layer will be living on a server with a connection to the Database (DB) API and a connection to the wearable.

Wearable
: The wearable will need to connect to the communication layer and process input and output.

The mock WMS and the REST API are insignificant parts of the implementation, therefore not a lot of thought was put into the decision on choosing these technologies was done out of curiosity for these technologies or being the most comfortable with them respectively.

5.2 Demo Scenario

The demo scenario explains the process that will be executed during the demo facility. Therefore describing the actions taken in detail, but ignoring how they will be executed, e.g. with or without a wearable. The original process can be seen in the appendix, figure A.1. For the actual scenario for the demo facility an activity diagram was created, that can be seen in figure 5.1.

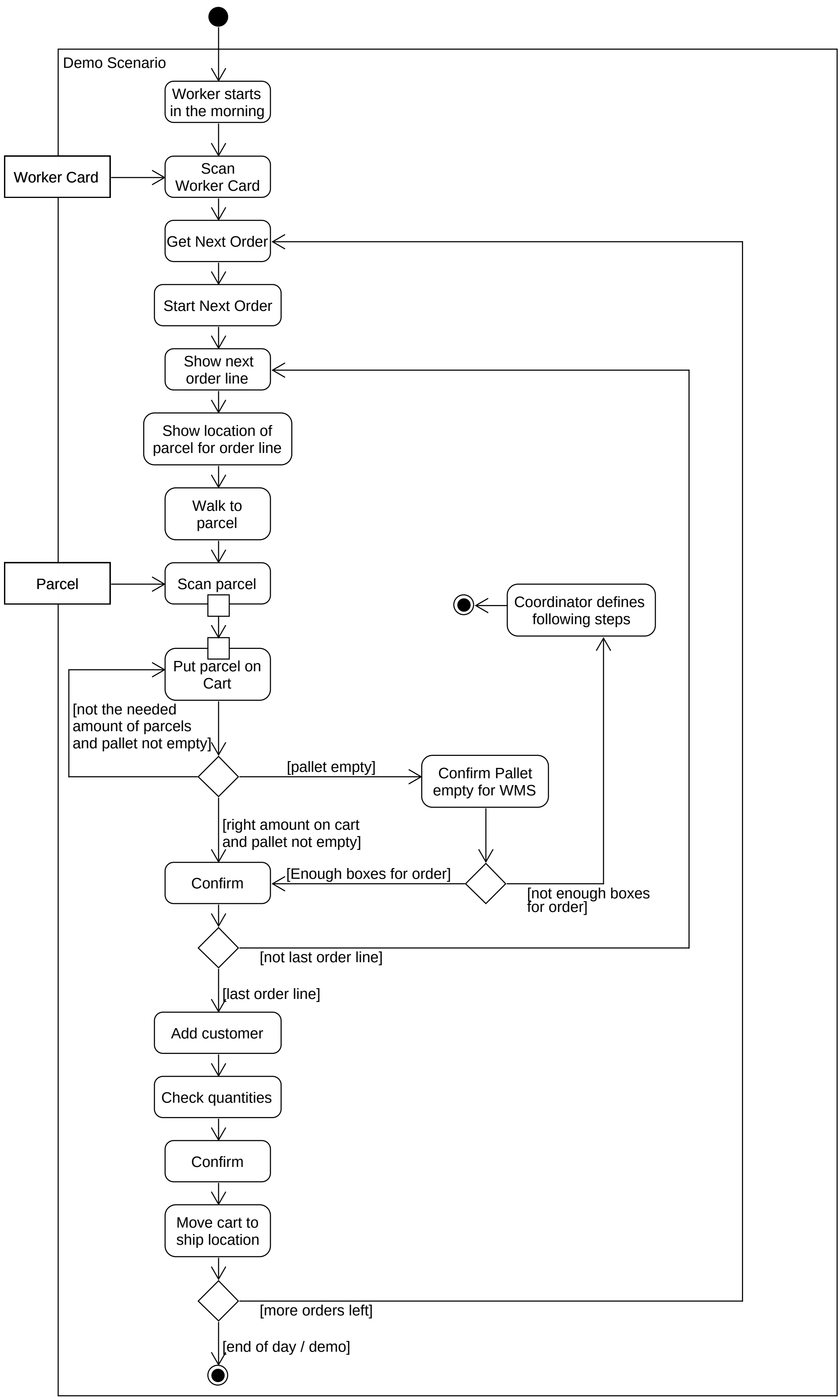

Figure 5.1: Activity Diagram Demo Scenario

This diagram was created for multiple reasons:

Readability
: The scale of the original process diagram made it hard to process. One goal was to create a more compact diagram.

Purpose
: The original diagram had a different purpose, it was supposed to model the process of one of the pilot companies in its entirety. The purpose of the activity diagram is to model a demo case, that does not need to show every single detail of the process in the first place.

Focus point
: The demo scenario focuses on the general tasks an order picking worker is doing, but is just focussing on the main points for this. The original diagram also includes the connections to the database and also includes tasks outside of the actual order picking.

add description of model

5.3 Design

The general design for the demo facility application will be based on the reference model described in chapter 4. But in the following sections the demo facility specific design will be elaborated further.

5.3.1 WMS

The Warehouse Management System consists of two parts, the database that is going to contain the data for the demo facility and the database connector. This also defines the interface, with which to connect to the WMS.

Database

Figure 5.2 shows the relations and fields in the database. It can be seen that the database just contains a small amount of information, due to being a demo, an actual WMS would contain a lot more data. The most important item in the model is the order, as that is the key piece, where most relations lead together. An order has a number that is connecting it to one or multiple workers that are working on them. Furthermore an order consists of multiple `OrderLines`. An `OrderLine` is describing the different lines that would appear on an order, that specify the item and the amount for an order. For a warehouse it is also important to add the pallet where to find the item and if the current line is already acknowledged or not. A pallet has a location in the warehouse and how much of that item are still available in the warehouse. The article corresponds to a name for the article number. Finally an order is ordered by a customer, a customer might have additional wishes for their orders and an address where that customer wants things delivered to.

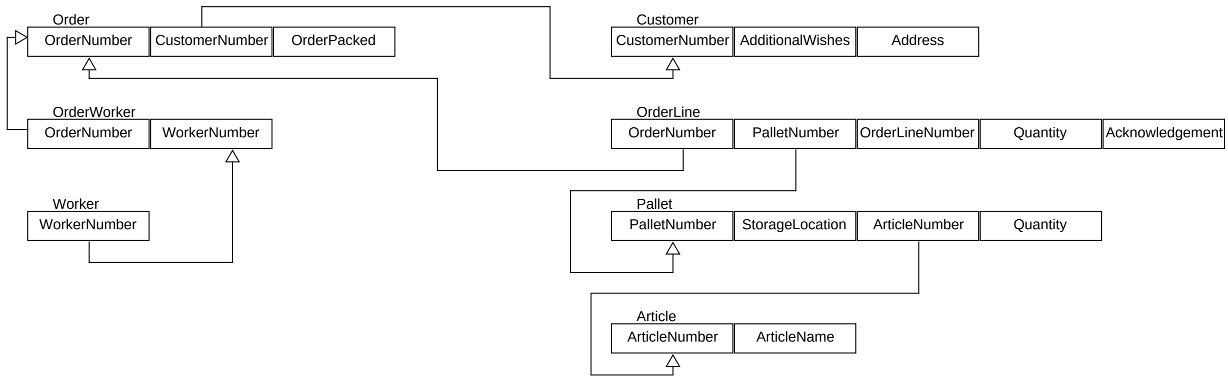

Figure 5.2: Relational Schema Warehouse Database

For an actual warehouse the customers attributes would change completely, as companies might have multiple addresses, therefore that might change for every order of that customer, making it easier to add an address field to the order and not the customer. Also a customer might want to add additional wishes just to a specific order or dependent on what might be ordered, therefore the additional wishes might also be moved to the order, but for a demo case, that is not executed at a pilot company, the model is sufficient.

Interface

The interface, that is exposed by the WMS, is a REST interface, for the demo case this means that the communication layer, explained in subsection 4.3.2, can be skipped, as the wearable chosen in section 3.1 is able to connect to a REST API directly. And for a demo case the communication layer is also not needed to further transform or do similar things with the data that is passing through it. The options available through the REST interface are the following:

NextOrder
Returns the order with the smallest number for a specific worker.

ConfirmOrderLine
Confirms that an order line in an order has been successfully been picked.

ConfirmOrder
Confirms that a complete order has been picked.

ResetDatabse
A function that exists purely to have an existing and repeatable demo case. This resets the database to a state it was in at the beginning, before the demo was executed.

5.3.2 Wearable | Application

Add information on how the wearable application works, maybe with web interface or direct implementation

add more sophisticated class diagram

5.4 Planning

In the future for this task, there will be multiple wearables tried out and then decided which wearable will be used for the demo facility, as described in section 3.2. Afterwards an application will be written for that wearable, to improve the process chosen in 3.1.

This application will at least include the functionalities to:

- Divide a given room in multiple sectors, to address a single part of the room with a given location string.
- Reset the case to allow multiple showcases of the demo.
- Allow scanning of IDs on parcels and pallets.
- Send confirmations to the WMS, that an action has been completed.

For the demo environment, an area will be rented that allows to place a rack in there with parcels. The parcels will be equipped with an ID that can be scanned with the chosen wearable.

When the implementation of the demo application is finished a second one could be started to show SME the differences between the possibilities different wearables give the user.

6 Conclusion

The reference model is the only current deliverable really created up to this point. This is due to the reference model being a crucial point of the research project. As the research goes on the reference model will be used as the artifact that is used to show developers on how a wearable system should look like in a logistics environment. Therefore the design went through multiple iterations to allow it to fit as many use-cases as possible, while reducing the need to change a lot of the design for each use-case.

The design of the reference model, even though a few communicational problems arose with what was actually expected from it, was successful and the involved parties are happy with how it turned out.

Further Planning

Now that the design of the reference model is at a stable state, it can be started to implement the parts of it that are not directly relying on the wearable itself. The wearables will be tested as they become available and a decision will be made depending on the results from that testing. Once that is done, the further creation of the demo facility will be able to planned accordingly and implementation of that can start.

When the creation of the demo facility is finished it is planned to invite SMEs to test out the wearables and how they could improve their processes.

Bibliography

Canders, Sascha (2017). “Development of a web application backend in a volatile environment”. Bachelor Thesis. Fontys Hogeschool Techniek en Logistiek. add page number.

Logwear (2017). *Order Picking Process Diagram*. [online internal].

Logwear.eu (2017). *LOGwear Arbeitspakete / Werkpakkete*. [online] Available at: `http://logwear.eu/`. [Accessed 25 March 2017].

Oasis-open.org (2017). *OASIS SOA Reference Model*. [online] Available at: `https://www.oasis-open.org/committees/soa-rm/faq.php`. [Accessed 22 March 2017].

Sander, Oliver (2017). “Development of a web application in a volatile environment”. Bachelor Thesis. Fontys Hogeschool Techniek en Logistiek. add page number.

Schwerdtfeger, Björn (2009). “Pick-by-Vision: Bringing HMD-based Augmented Reality into the Warehouse”. PhD thesis. Technische Universität München. add page number.

Umlet.com (2017). *UMLet Website*. [online] Available at: `http://www.umlet.com/`. [Accessed 25 March 2017].

Appendices

A.1 Order Picking Process

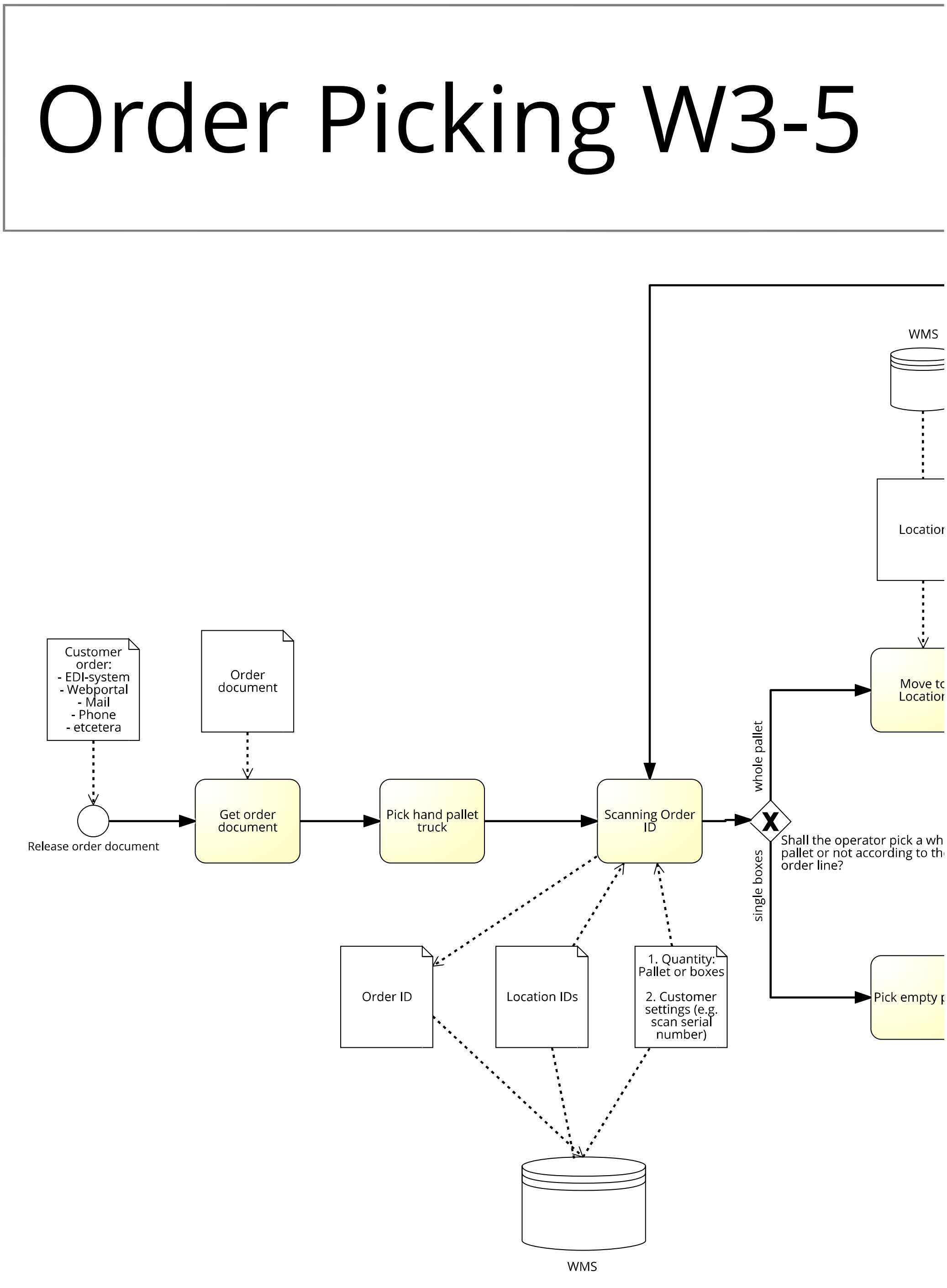

Figure A.1: Order Picking Process Diagram (Logwear, 2017)

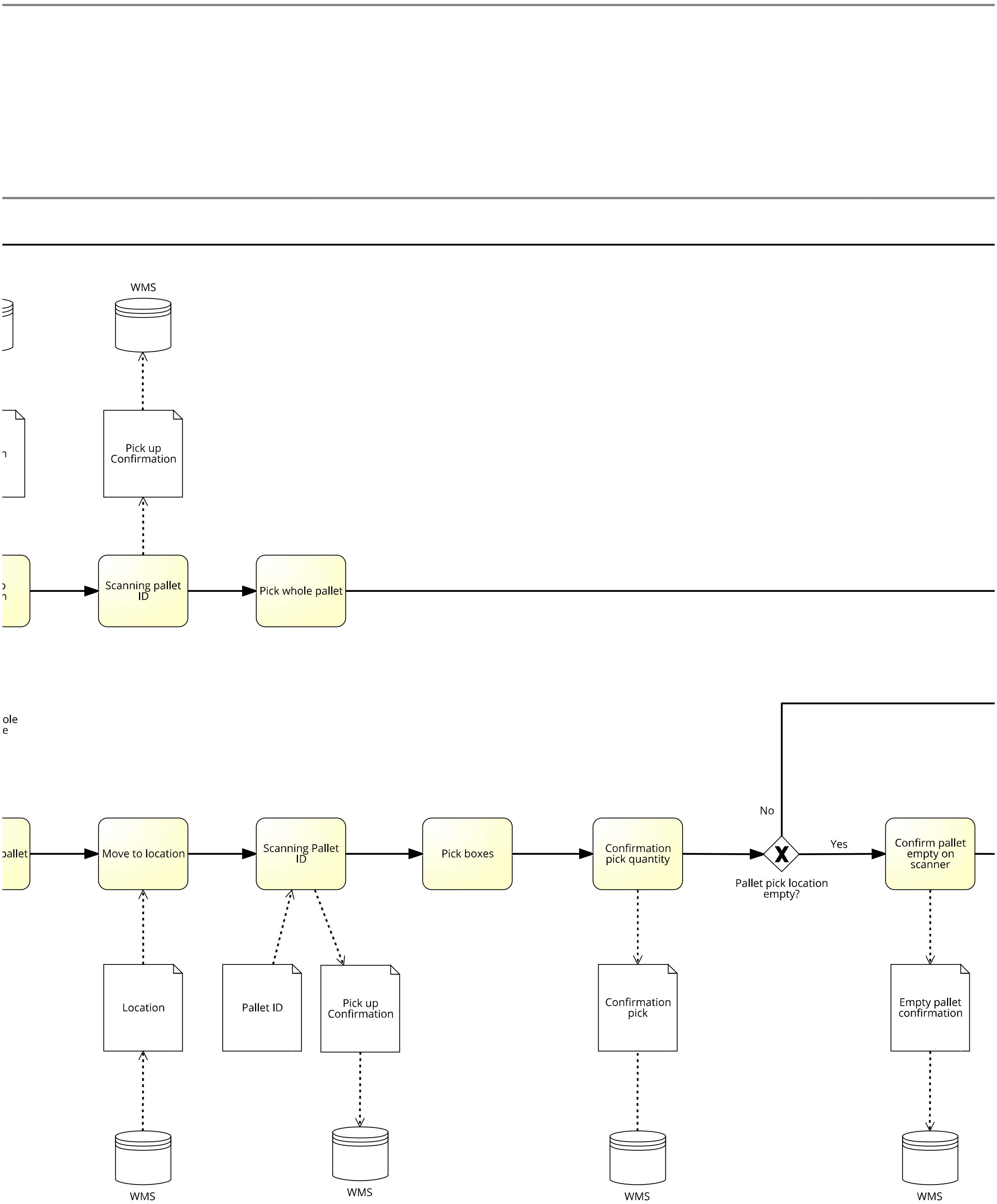

Figure A.1: Order Picking Process Diagram (Logwear, 2017)

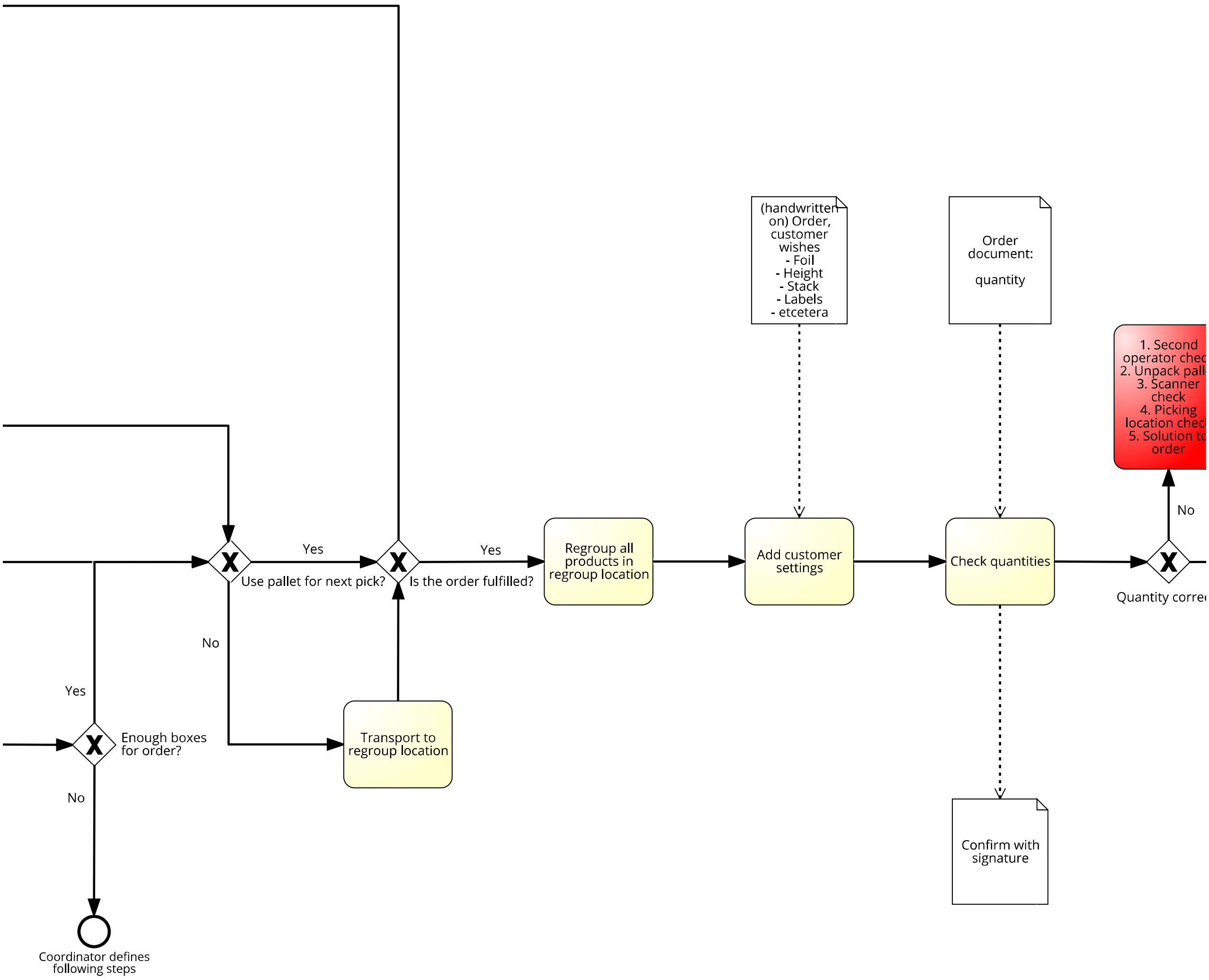

Figure A.1: Order Picking Process Diagram (Logwear, 2017)

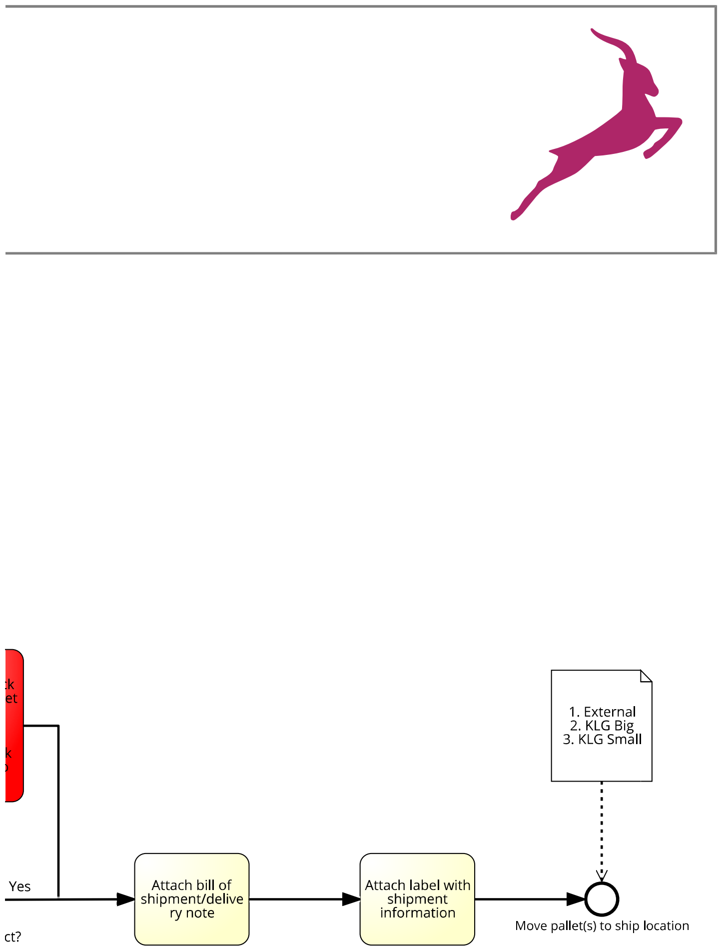

Figure A.1: Order Picking Process Diagram (Logwear, 2017)

A.2 Use Cases Reference Model

Use Case	**Get Order (Voice)**		
Code	UC-W1		
Package	Wearable		
Actors	Picking Worker		
Description	A picking worker equipped with a wearable is using a voice command to get the next order displayed.		
Precondition(s)	The wearable has to be equipped with an Input Interface accepting voice and and an Output Interface that is able to return information to the user.		
Scenario	1. The Picking worker gives the voice command "Next Order". 2. The Wearable asks the WMS for the next order for the specific worker. 3. The WMS sends the Data to the Wearable 4. The Wearable displays the Data to the order picker.		
Extensions	-		
Exceptions	1.1 The Voice command could not be properly understood. The Wearable does nothing. 1.2 The Order Picker is already on an Order and that one is unfinished, the command is ignored. 1.3 Another Voice command is understood, that one is executed. 3.1 The WMS did not find a next order. The order picker is informed about that. 4.1 There is an error in the format that was received. The data that could be understood is still displayed and the order picker is informed, that there might be an error with the order.		
Result	The Order Picker has received the next order.		
Version	1.0	Author	LUR

Figure A.2: Use Case: Get Order(Voice)

Use Case	**Order Confirmation**		
Code	UC-W2		
Package	Wearable		
Actors	Picking Worker		
Description	A picking worker equipped with a wearable is using an input interface to confirm an order.		
Precondition(s)	The order picker has finished picking the order and wants to confirm that everything is correct.		
Scenario	1. The order picker uses an input interface on the Wearable to confirm the Order. 2. The wearable is processing the request. 3. The wearable is sending the Confirmation to the WMS. 4. The WMS sends, that the confirmation was successfully received. 5. The wearable displays the user, that the confirmation was received by the WMS.		
Extensions	1.1 The Wearable is able to check if the confirmation is allowed to be send. See UC-W3.		
Exceptions	3.1 The Wearable got an error when processing the request. The order picker is informed about the cause of the error and can try to fix it. 4.1 The WMS does not answer. The order picker is informed about that and can try contacting the IT.		
Result	The Order Picker has confirmed his order.		
Version	1.1	Author	LUR

Figure A.3: Use Case: Order Confirmation

Use Case	**Order Control**		
Code	UC-W3		
Package	Wearable		
Actors	Picking Worker		
Description	A picking worker is in the process of picking his order.		
Precondition(s)	The wearable has a vision interface or is in another way able to control what the order picker is doing. Furthermore the order picker is currently in the process of picking an order and is picking a specific item of that order.		
Scenario	1. The order picker scans an item. 2. The wearable checks what is being packed. 3. The order picker is putting the item on his hand pallet truck. 4. The wearable is counting the amount of items put onto the hand pallet truck. 5. The order picker scans a new item. 6. The wearable detects a new item and ends the counting process for the last item. 7. The wearable informs the order picker that everything went correctly with the last item.		
Extensions	6.1 When the order has no next item, the worker tries to confirm the order and the wearable can then proceed to check if the amount of picked parcels was correct.		
Exceptions	7.1 The order picker could have counted wrongly and the wearable is informing him about it. After the order picker has checked the quantity on the Truck, go back to 1. with the item started with. 7.2 The wearable could have counted wrongly and the wearable is informing the order picker as if he counted wrong. The order picker can check the hand pallet truck for the item and confirm the right quantity. 7.3 The order picker could have picked the wrong quantity and the wearable could have counted wrong, resulting in the wearable saying the order picker has picked the right amount. An error is made.		
Result	The Order Picker completed a part of the order and the wearable confirmed the quantity of that part.		
Version	1.1	Author	LUR

Figure A.4: Use Case: Order Control