Software requirements specification – Team 01

Inhaltsverzeichnis

Introduction 2

Purpose 2

Stakeholders 3

Definitions 3

System overview 3

References 3

Overall description 4

Use cases 4

Actor characteristics 4

Company/Logisticians: 4

Drivers: 4

Customers: 4

EPK 5

Objects 6

Specific requirements 6

User Interfaces 6

Hardware Interfaces 6

Software Interfaces 6

Communication Interfaces 6

Functional requirements 6

User Class 1 – The Logistician 6

User Class 2 – The Driver 6

User Class 3 – The customer 6

Non-functional requirements (external, performance, etc.) 6

Design constraints 6

Software System attributes 6

# Introduction

## Purpose

The goal of this software is to provide an application to simplify the process from ordering to delivery. In the current situation everything is done by hand from each employee in the process line, leading to an excessive amount of paper work. With an automated system the company wants to improve the efficiency of their logistics.

This Software Requirements Specification document serves as a guideline for the project and gives a detailed overview of the functionalities of the software. Its aim is to capture the needs of the company *RCP* and establish a common understanding of the functional and non-functional requirements of the project.



## Stakeholders

The stakeholders are Manuel Leuenberger of the company *RCP* and its employees including the logisticians and drivers.

## Definitions

**Logistician**: employee who handles incoming orders, prepares tours and assigns a driver for each tour.

**Driver**: carries out the tours, which are assigned to him, and delivers the products to the customer.

**Customer**: company to which the products are delivered to and which is able to accept or reject the delivery.

**Tour**: is a composition of different orders assembled by a logistician.

**Accepted delivery**: the order is successfully delivery and marked as accepted by the driver.

**Rejected delivery:** the order could not be successfully delivery due to a mistake in the delivery or the unavailability of the customer. It is then put back into the unassigned orders.

## System overview

The system should:

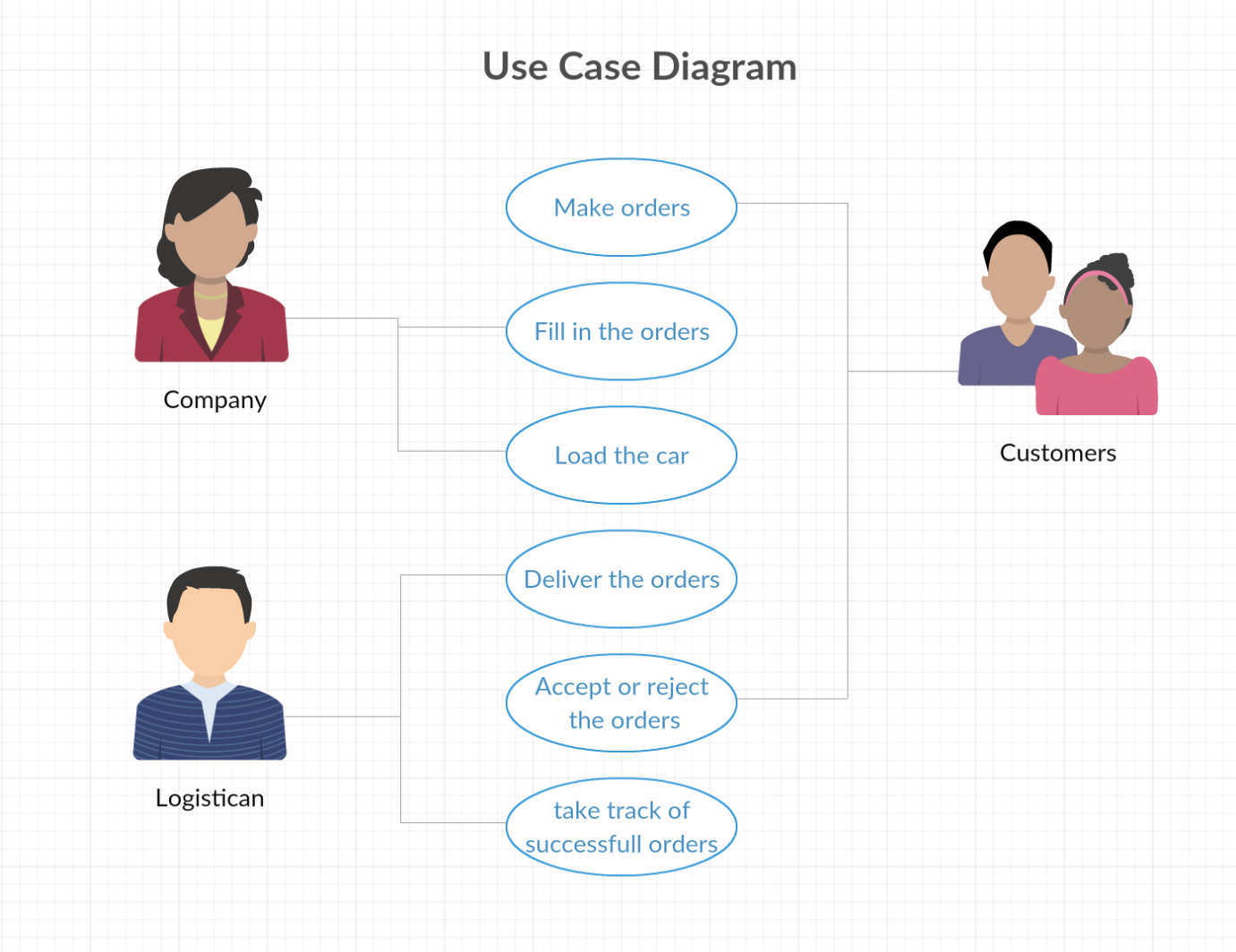
* Save the logisticians in the database, they have the access to add, change and look up things
* Save the drivers in the database, they can just look up things and add the state of the order
* Save the customers with their addresses in the database
* Save the orders from the client in a database
* Prepare tours with the different addresses of the clients
* Show the tours for the logisticians on a mobile device
* Let the drivers access the orders for changing the state of the order (delivered, rejected, not there)

## References

„here come what literature we use“

# Overall description

## Use cases



**Name:** Log into the system.

**Description:** A driver or logistician, which is already registered visits the website and logs in with his credentials.

**Preconditions:** 1. User has access to the Internet

2. User has an account on the website

3. User knows his e-mail address and password

**Event Flow:** 1. User types in e-mail address

2. User types in password

3. User clicks login button. In case the credentials are correct, he is led to the driver or logistician homepage, depending on which he has access to. In case either the e-mail address or the password is incorrect, the user is requested to verify.

**Name:** Log out of the system.

**Description:** A driver or logistician, who is logged in, logs out of the system.

**Precondition:** 1. User has access to the Internet

2. User has an account on the website

3. User knows his e-mail address and password

4. User is logged in

**Event Flow:** 1. User sees Log out button on the top right of the website

2. User clicks Log out button and is taken back to the Login page

## Actor characteristics

### Company/Logisticians:

The company that orders the program is taking the orders of the customers. They are making the loading lists and filling the trucks with the ordered goods.

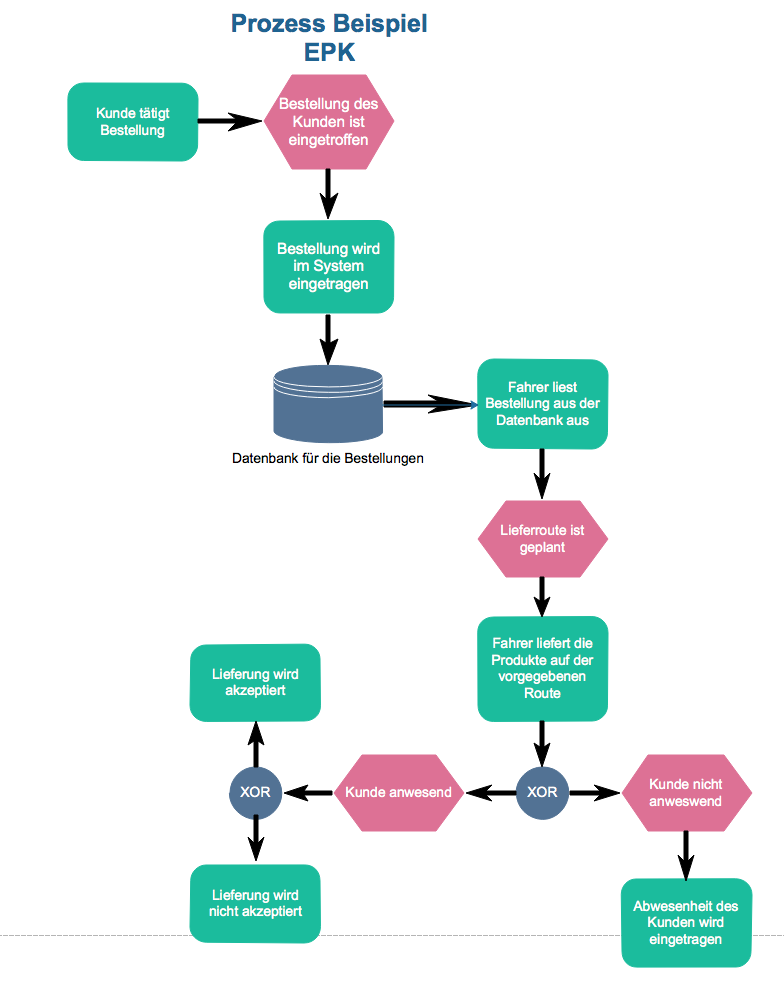
### Drivers:

They are taking the delivery lists and drive the ordered goods to the customers place. They have to take track of the accepted and rejected deliveries.

### Customers:

They order the goods from the company. They get the goods from the logistician and can accept or reject them when they are there.

## EPK



# Objects

# Specific requirements

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communication Interfaces

## Functional requirements

### User Class 1 – The Logistician

### User Class 2 – The Driver

### User Class 3 – The Customer

## Non-functional requirements (external, performance, etc.)

## Design constraints

## Software System attributes