

# Software requirements specification (SRS)

Christof Sy, Michael Trittibach, Michael Brunner, Manuela Eschler

Introduction to Software Engineering

Computer Science  
University Bern

Version: 1.1

October 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Stakeholders . . . . .	3
1.3	Definitions . . . . .	3
1.4	System Overview . . . . .	3
1.5	References . . . . .	3
1.6	Open Questions . . . . .	3
1.7	Answered Questions . . . . .	4
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	Use Cases . . . . .	5
2.1.1	Use cases of the logistician . . . . .	5
2.1.2	Use cases of the driver . . . . .	5
2.1.3	Use cases of the user admin . . . . .	5
2.1.4	Use cases of the system . . . . .	6
2.2	Actor Characteristics . . . . .	6
2.2.1	Logistician . . . . .	6
2.2.2	Driver . . . . .	6
2.2.3	User Admin . . . . .	6
2.3	Open Questions . . . . .	6
2.4	Nice-to-have List . . . . .	6
<b>3</b>	<b>Specific Requirements</b>	<b>7</b>
3.1	Functional Requirements . . . . .	7
3.2	Non-functional Requirements . . . . .	7
3.3	Open Questions . . . . .	7
3.4	Answered Questions . . . . .	7

# 1 Introduction

## 1.1 Purpose

The System will inquire the following purposes:

- Reduce work/staff  
The System will reduce time to do the same work. Less persons can do the same job with the System.
- Get overview of all orders and drivers  
The logistician will have a good overview of all the orders, drivers and deliveries.
- Always available and up to date  
The informations on the system are always available and up to date.
- Analyze for improvement  
The estimated time for the deliveries and the actual time used will differ. The system allows to analyze this difference for improvements.

## 1.2 Stakeholders

The stakeholders represent the groups of people who will use the system. The system will serve their purposes in the best way possible.

- **Driver**  
The driver makes the deliveries. He needs at any time access to his current and future deliveries and their informations.
- **Logistician**  
The logistician organizes the tours for the drivers. He keeps the overview and reschedules failed deliveries.
- **User Admin**  
The User admin manages all users.

## 1.3 Definitions

No definitions required yet.

## 1.4 System Overview

The new System's overall purpose is to manage logistics. It assists the stakeholders driver and logistician in the best possible way.

**The logistician** has a overview of all the pending orders at every time. Out of all the pending orders the system supports him organizing tours for every driver. With this system the logistician is able to adjust and reorganize the tours if necessary with little effort.

The system supports analyzing completed tours for improvement.

**The driver** has an overview as well as details of his few next tours. He can update the status of every delivery in real-time.

The driver can take a look at his tours on a desktop as well as on a mobile. **The user admin** administrates all users on the system.

## 1.5 References

No references required yet.

## 1.6 Open Questions

No open questions for now.

## 1.7 Answered Questions

1. Is a forth stakeholder necessary? Like a customer? Where are the orders from?  
A customer is currently not necessary, maybe a nice thing to have. The orders are coming from our kiosk.
2. If there is a customer, can he follow his delivery?  
He would definitely get a notification if the status of his delivery changes, but for him to see where his package is at the moment would also be a nice thing to have.
3. Login, private link or open source?  
No need to worry about that, we'll only use it locally.

## **2 Overall Description**

### **2.1 Use Cases**

#### **2.1.1 Use cases of the logistician**

The logistician can login to the website with his username and password. Here he has a broad overview of all deliveries and their information and he can assemble the open ones to tours. He then can assign tours to the drivers. After a tour is finished, the logistician can evaluate the estimated against the used time. When a delivery was unsuccessful, he then can reschedule these to different future tours.

1. Logistician login to website  
The logistician opens the website with a browser and logs in with username and password to the backend.
2. Logistician organizes tours  
The logistician has overview of all deliveries and their informations. He assembles the deliveries to tours.
3. Logistician assigns tours to drivers  
The logistician assigns the tour to a driver.
4. Logistician evaluates tours  
The logistician evaluates for every tour the estimated vs the used time.
5. Logistician reschedules failed deliveries  
The logistician adds failed deliveries to future tours.

#### **2.1.2 Use cases of the driver**

Just like the logistician, the driver can login to the website with his username and password. Here he has an overview of his tours with details of every delivery. He can change the status of a delivery to passed or failed and also adds the used time for the tour.

1. Driver login to website  
The driver opens the website with a browser and logs in with username and password to the frontend.
2. Driver has overview of his tours  
The driver has overview of his tours with details of every delivery.
3. Driver sets status of delivery  
The driver changes the status of a delivery to passed or failed.
4. Driver adds used time for tour  
The driver adds the used time for the tour.

#### **2.1.3 Use cases of the user admin**

The user admin logs in to the website like the others with a username and a password. He can add new users to the system and can set their settings.

1. user admin login to website  
The user admin opens the website with a browser and logs in with username and password to the backend.
2. User admin creates users on system  
The User admin adds new users to the system and set the settings right.

#### **2.1.4 Use cases of the system**

The system notifies the logisticians when a delivery status changes to failed.

1. System notifies logistician when delivery status is failed The system sends a notification to the logistician when the status of a delivery changes to failed.

### **2.2 Actor Characteristics**

#### **2.2.1 Logistician**

The logistician would like to:

- Organize tours with help from the system
- Improve tours with little effort by evaluating past tours
- Reschedule tours without additional paperwork
- Have an Overview of the status of the deliveries

#### **2.2.2 Driver**

The driver would like to:

- Have an overview of his upcoming tours
- Update of the status of the deliveries immediately

#### **2.2.3 User Admin**

The user admin would like to:

- Add new users to the system
- Remove users from the system
- Modify users on the system

### **2.3 Open Questions**

1. These are our initial recommendations to accomplish your request. Open for discussion.

### **2.4 Nice-to-have List**

This is a list for things to further update the system, once the important parts are finished:

1. Users can have multiple roles.
2. Drivers can transfer tours to other drivers, when they are not available at the time.
3. Users can reset their password without the help from a user admin.
4. Costumers can follow the status and place of their package.
5. The System can calculate the time the tours will take.

## 3 Specific Requirements

### 3.1 Functional Requirements

Backend:

- User management: Add, modify and remove users  
Every user needs a username , a password and a status (logistician, driver, user admin or inactive).  
Inactive users cant log in to the system.
- Tour management: Add, modify and assign tours  
Logisticians have to be able to combine deliveries to a tour and assign them to a driver.  
Drivers have to be able to watch their upcoming tours and details of the tour. Drivers have to be able to report the used time of the tour.  
Logisticians have to be able to get a overview of the statistics of the passed tours.
- Delivery management: Add, update and modify deliveries  
Logisticians have to be able to get a overview of the deliveries witch are not already assigned to a tour or witch are failed.  
Drivers have to be able to set the status of a delivery to passed or failed.
- Report management: Compare estimated and used time of tours  
The system have to compare estimated and used time of tours, how many deliveries and how much of them passed or failed.

Frontend:

- Individual tour overview: Present tour and the assigned deliveries incl. details  
Drivers have to get a overview of their upcoming tours directly after logging in to the system. Before they start the tour the have to tell it to the system and at the end of the tour they have to tell it to the system.
- Delivery Status management: Update status of delivery  
After every passed or failed delivery the driver have to report the status of the delivery to the system.

### 3.2 Non-functional Requirements

- The web application has responsive design

### 3.3 Open Questions

No Questions for now.

### 3.4 Answered Questions

1. How many user will be on this system?  
Not really many, we are a small company.
2. If a delivery fails, should the driver be able to try it again on the same tour?  
Yes, he can go by another time.
3. Where is the server located?  
That's not important.
4. How many times a delivery can fail? What happens after it fails to much?  
Maybe like 3 or 4 times, after that we'll send the delivery back to the sender.
5. How does the paperwork look like? take this as template?  
Information about the customer and the delivery.

About the customer:

- Company
- Contact person
- Address
- Phone number
- E-mail
- Delivery acceptance window

About the delivery:

- Height length
- Width length
- Side length
- Weight
- Information of the content in the package (fragile, dangerous...)
- Comments