SOFTWARE REQUIREMENTS SPECIFICATION

for

Logistics Management Tool

Release 0.1

Version 1.0approved

Prepared by Dave Meier, Olivier Ulrich,

Contents

Revision History

Name	Date	Reason For Changes	Version
1	02.01.2017	Initial Release	1.0

1 Introduction

1.1 Purpose

The purpose of this project is to create a piece of software which allows logisticians and drivers to plan, drive and deliver orders faster and more optimized than the current solution with a lot of analog papers and calendars. Also the plans should be flexible, dynamic and adjustable at any given time as well as accessible to all relevant participants.

1.2 Stakeholders

The stakeholders are the the team of the ESE-Assistants (.,.,.,.) and ESE-Team3 (Dave Meier, Olivier Ulrich, Luca Rolshoven, Julian Weyermann)

1.3 Definitions

- **Tour:** Route containing one or more deliveries
- **Workload:** The total amount of deliveries
- **Unsucessful delivery:** Deliver that could not be handed finished due to the customer rejecting it or some other irregularity

1.4 System Overview

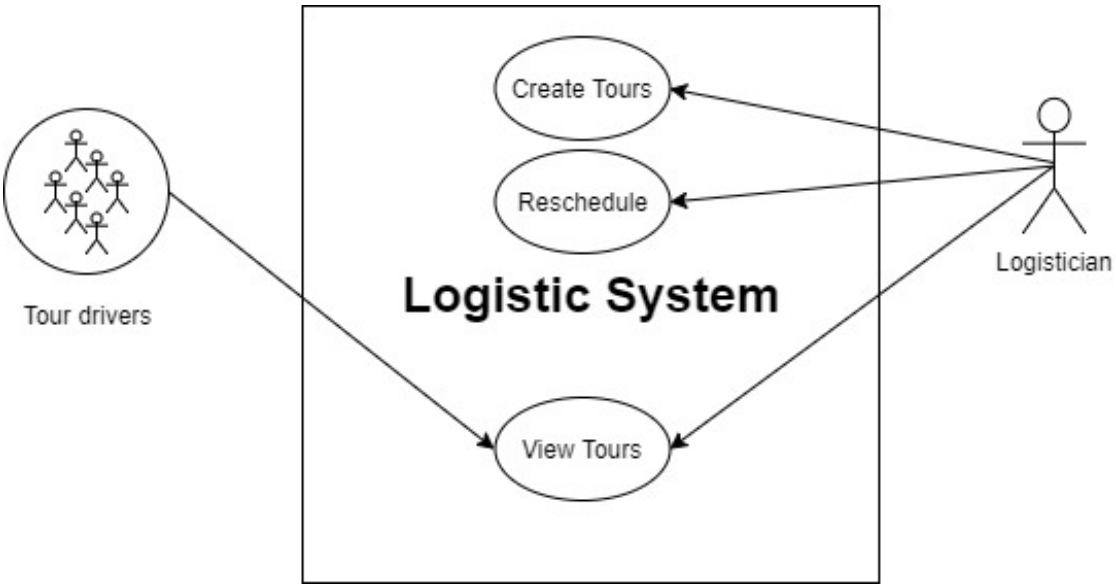
The Users are Logisticians and Tourdrivers. Maybe also an Admin who is able to view all tours and reassign and modify them. A tour consist of deliveries and tours (or deliveries?!?!?) are introduced by a logistician. (Does a Tourdriver need to modify a tour?). Tourdrivers can access those tours and confirm or reject single deliveries. We have a Pool of deliveries (non delivered-deliveries) of which a certain number can be picked and packed to a tour. If a delivery of a tour fails, the tour is considered as sucessfull and finished, but the deliveries which failed fall back into the pool of non delivered-deliveries. Deliveries are assembled to a tour by their geolocation. Either the tours are fixed and the deliveries are assigned to the tour or the tours are dynamically assembled together by the geodata of the deliveries. A Delivery has additionally at least the date of the creation of the deilvery. Maybe there is other data needed: Delivery priority, number of failed delivery attempts, weight (to avoid overload of vehicles), if its back in the warehouse or if its on tour etc.

1.5 References

Delivery Systems of e.g. UPS, DPD and maybe other logistics service providers

2 Overall Description

2.1 Use Cases



-
- Who decides which driver gets which tour? -¿ - Logisticians assign tours to a driver -¿ - Logisticians create tours and drivers pick the ones they appeal them the most (first come first serve)
- The tour drivers have a tour start/stop button on their personal account to monitor the time of the tour

2.2 Actor Characteristics

Tour drivers only need the ability to Start/Stop(/Pause) tours and confirm/reject deliveries, while Logisticians need to be able to modify (and create/delete) tours and deliveries.

3 Specific requirements

3.1 Functional requirements

The logistic system should be able to automatically assemble some tours in a way that the total amount of logisticians is minimized while processing the same workload. It should provide an interface that allows the logisticians to assemble, change or cancel tours and there should be an interface accessible via mobile phones so that the drivers can see their daily tours as well as an overview over the coming tours during the next few days. Furthermore the system distinguishes between single deliveries and whole tours and it provides mechanisms to mark deliveries as successful or unsuccessful. Moreover it offers a tracking mechanisms that tells the user if the delivery took longer than expected, so the delay can be handled. Also the system handles permissions for drivers/logisticians as expected and as claimed by the Stakeholders.

3.2 Non-functional requirements

- Mobilefriendliness
- Real-time
- Possibly Multilanguage?!
- Java, Maven, Spring, MySql