Team 7

Ratheeban Rajakumar
Yves Chapuis
Roman Alonzo
Roland Tschendel
Michael Monteiro

Software Requirements Specification

Document

Version: (1.8.8) **Date: (12/12/17)**

Content

1. Introduction

1.1 Purpose

The purpose of this SRS is to describe the requirements for the logistics tool programmed for the aniTrans company during the ESE course at the University of Berne in the autumn-semester 2017.
It is intended for aniTrans and team 7.

1.2 Scope

Our app will provide the logisticians of aniTrans the means to plan tours for their drivers. The drivers will be able to see their tours. The driver can mark his or her tours as ‘delivered’ or ‘undelivered’ if the recipient isn’t home and add a comment.
The purpose is to make life easier for aniTrans, which have up to now been writing everything on paper.

1.3 System Overview

The SRS will contain first a general description of the project and then the specifications, which are intended mainly for the team and not the customer. There we will make technical definitions which are only relevant to us.

1.4 Stakeholders

The main stakeholder of aniTrans for this app is the managing director.

1.5 Definitions

- aniTrans: Animal Transportation company
- Spring Framework: a programming and configuration model for modern Java-based enterprise applications
- SRS: software requirements specification

2. The Overall Description

The application will be a web-application to manage the logistics of aniTrans, which is a company that transports animals.
After logging in, the logisticians will be able to plan tours for their drivers. The tours can be sorted by status (delivered & undelivered). Details of the order incl. google maps directions are provided on a separate page, which is accessible through a button.

A vehicles list gives an overview of all trucks which are available. Vehicles have an image, max. weight, length and width. Single vehicles as well as vehicle types can be added and deleted.
If a tour/order is created, it is possible to assign a vehicle. Take the vehicle out of the vehicle list (i.e. from the available vehicles). A vehicle may only be used on one tour per day. It is also possible to see details of every order/tour incl. google maps directions from the aniTrans headquarters at Hochschulstrasse 6, 3012 Bern to the starting point to the delivery point.
In order to provide a good usability of the application it should be possible to close or expand tables on the website.

The drivers will be able to create an account and log in to see their tours. Once on a tour they can mark individual deliveries as delivered or undelivered and add a comment.

Regular users of the website can access the homepage, where they're provided with a short list of information about aniTrans.

2.1 Product Perspective

2.1.2 System Interfaces

This will be a self-contained web-application, so the interface will consist of a graphical website. The website will contain images and texts to provide information to the user. To interact with the application there will be forms for login, creating and managing orders and tours and adding and deleting vehicles and drivers.

2.1.3 User Interfaces

The webpage consists of the following pages, forms or tables:
- Add/edit/view orders
- Change order status
- View tours
- View and delete drivers
- Register and login
- View/add/remove vehicles
- Add vehicle types
- Homepage

2.1.5 Software Interfaces

The web app will use a MySQL database to store information.

A web server needs to be installed to run the website.
The customer needs a web browser.

2.1.6 Communications Interfaces

Thymeleaf is used to communicate between html forms and sites and the java code of the web-application running behind the website.
Spring Data JPA is used to communicate with the MySQL database.

2.1.7 Memory Constraints

Standard Business PC requirements.

2.2 Design Constraints

2.2.1 Operations

Version 2

We must be able to interface with any html browser.

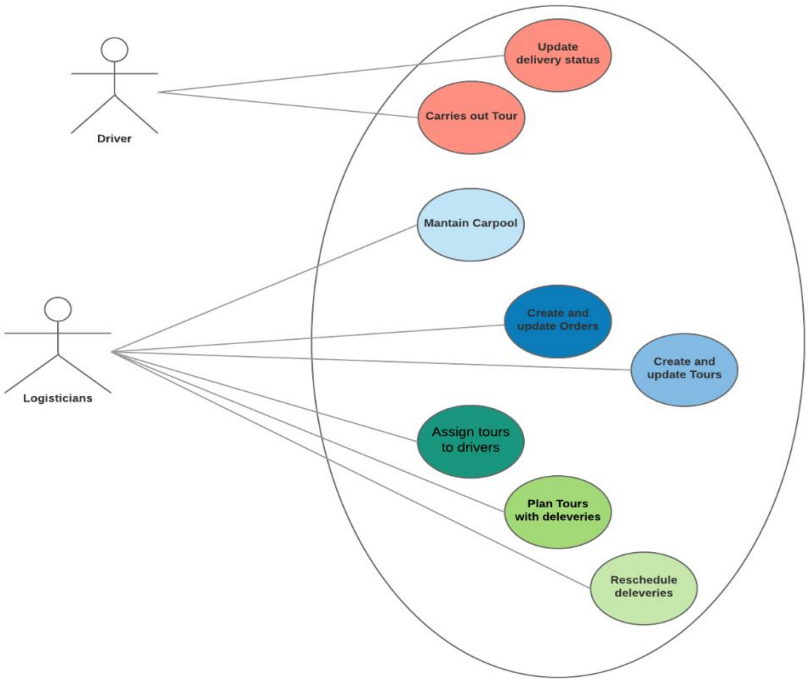
Out of scope

- The new version of the spreadsheet must be able to access data from the previous two versions.
- The product shall be able to be installed by an untrained user without recourse to separately-printed instructions

2.2.2 Standards Compliance

We don't have any standards we have to conform to.

2.3 Product Functions



- Logisticians
  - Log in
  - Plan tours
  - See which deliveries were delivered and which were not
  - Assign tours to drivers
  - Add and remove vehicles or vehicle types
  - View and delete drivers
- Drivers
  - Create an account and log in.
  - See the tours incl. google maps and change the status of tours
- Regular users
  - View homepage
  -

2.4 User Characteristics

General Preconditions

User has access to the internet and a working computer or smartphone.

User Characteristics

Standard Users: Ability to read English, able to register an account, able to navigate web-pages.

Logisticians: In addition to Standard User capabilities, they need to fill out forms, and manage a database through the graphical interface of the website.

Driver (standard) **Use Cases**

Description: Can see order-specific data, and can change the status of these orders.
Cannot see Orders, Vehicles or Drivers pages.

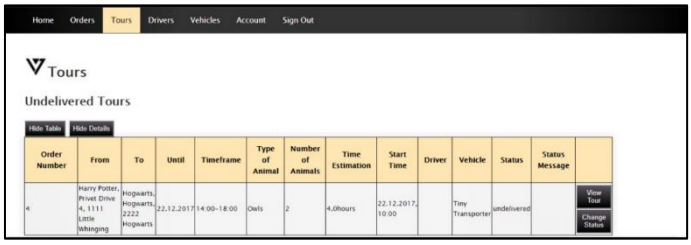
Pre-condition: Should be logged in as a driver.

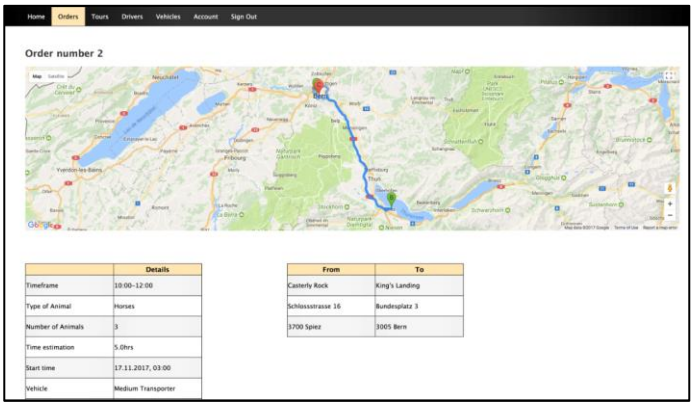
Post-condition: Order status changed.

Main scenario:

- Logs in.
- Navigates to tours.

- Reads tour information and carries it out.





- Depending on if the animals are delivered or not, updates the status to „delivered“ or „undelivered“.

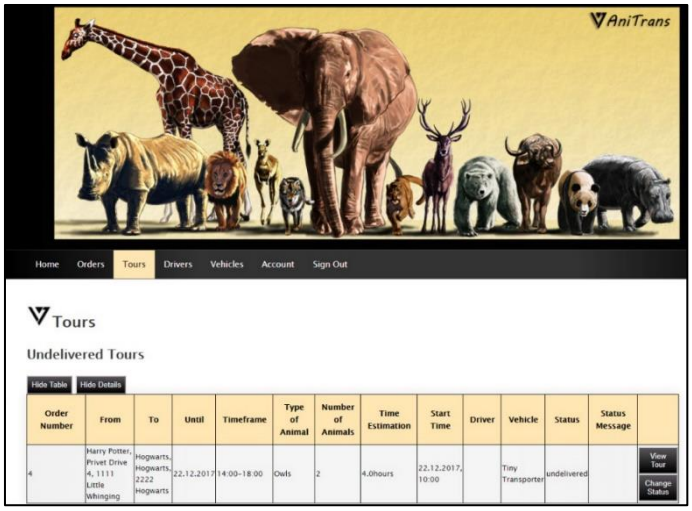
Logistician Use Cases:

Description: Can create orders and fill out required fields.
Can save orders and look into status changes made by drivers.
Can also delete orders, see and delete users, and see, add and delete vehicles

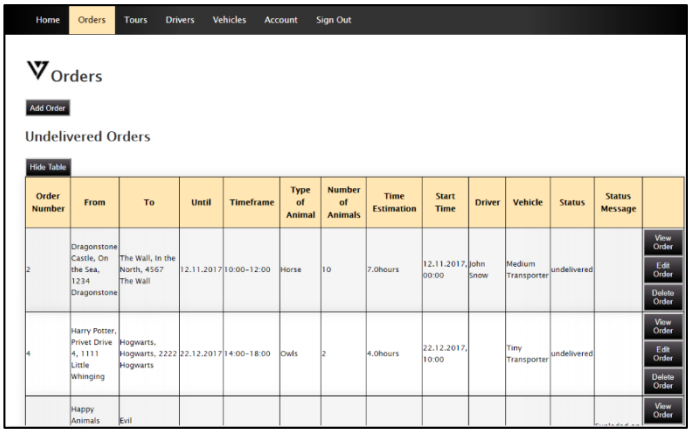
Pre-condition: Logged in as logistician.

Post-condition: New orders made, orders deleted, completed orders confirmed.

Main scenario:



- Navigates to Orders.

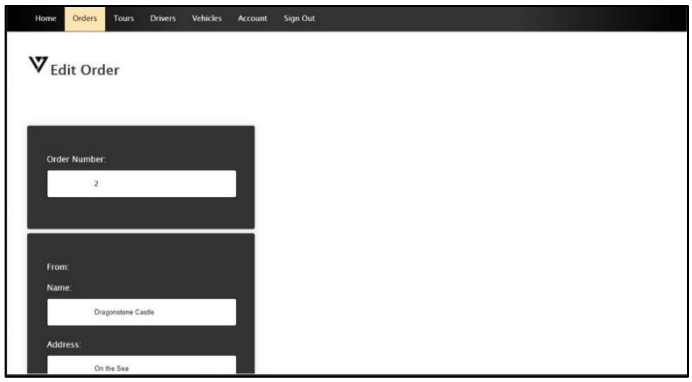


- Sees a list of pending orders as well as a create new order tab.

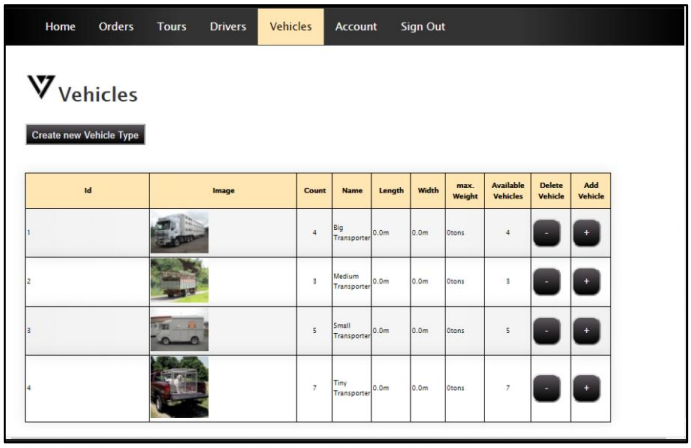
- If new order tab is selected, an empty form is displayed.

  If the form is filled out and confirm is clicked at the end, the order is saved and made visible for the drivers.

- Makes changes to existing orders.



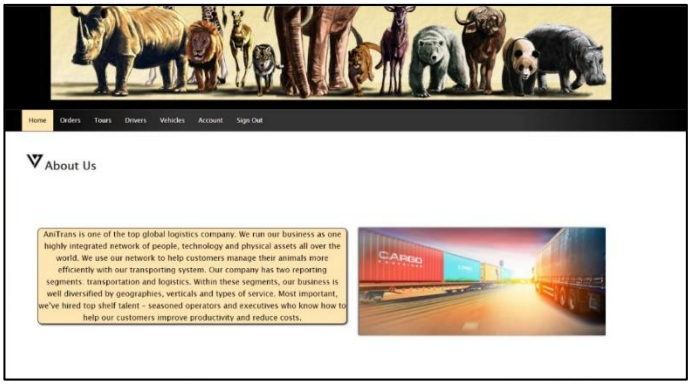
- Sees and deletes users.



- See, adds and deletes vehicles.

Non-registered User Use Cases:

User is unable to see orders, users, vehicles and drivers.
User can see the homepage to get information on aniTrans.



2.5 Constraints assumptions and dependencies

Version 1

- The product shall prevent incorrect data from being introduced.
- The product shall not be offensive to religious or ethnic groups
- The product shall make all functionality available to the managing director.

Version 2

- The product shall protect itself from intentional abuse.
- The product shall make its user aware of its information practices before collecting data from them.
- The product shall comply with logistics industry standards**.**

Out of scope

- The product shall be able to distinguish between French, Italian and British road numbering systems.

3. Specific requirements

3.1 External Interface Requirements

The application runs inside a browser. It should work with any browser but has only been tested it in Safari and Google Chrome so far.

3.2 Functional Requirements

- Adding/deleting/editing/viewing orders with the specified attributes (from, to, until, timeframe, type of animal, number of animals, time estimation, start time, driver, vehicle, status, status message) (only admin).
- Viewing orders and changing their status (drivers and admin).
- Viewing drivers/users (just the admin).
- Viewing vehicles, adding vehicle types and adding/removing single vehicle (only the admin).
- Version 2: multiple trucks may be added to an order.

- Registering, login and logout (anyone).
- Accessing the homepage (anyone).

4. Non-Functional Requirements

4.0 User Interface

- The user interface of the system should be designed in a way to make the systems functions accessible to most users without prior learning or training.
- The user interface should be intuitive and easy to use.

- The language of the frontend is English

4.1 Performance Requirements

The performance is good enough to make the response time acceptable to the user.

4.2 Logical Database Requirements

- The order table contains addresses (this is a foreign key relationship).
- The order table contains drivers and vehicles, but this is not a foreign key relationship, because we want the name of the driver/vehicle to stay even if the driver/vehicle is deleted. That way the user can see who the order was assigned to before that person was fired.
- The NewOrder, EditedOrder and NewUser tables never contain any entries. They only exist because their respective objects are used to transfer data from the html forms to the Java code. This is done because in the form the elements of an address (name, street, zip-code and town) are all single variables, whereas in the database it's one Address object. The java code then converts the NewOrder, EditedOrder and NewUser objects into AniOrder, User and Address objects.

1 address

Creation: Nov 16, 2017 at 09:57 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
name	varchar(255)		No					
plz	int(11)		No					
street	varchar(255)		No					
town	varchar(255)		No					

2 ani_order

Creation: Nov 16, 2017 at 09:57 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
driver	varchar(255)		Yes	NULL				
number_of_animals	int(11)		Yes	NULL				
order_status	varchar(255)		Yes	NULL				
start_time	datetime		Yes	NULL				
status_message	varchar(255)		Yes	NULL				
time_estimation	float		No					
timeframe	varchar(255)		Yes	NULL				
type_of_animal	varchar(255)		Yes	NULL				
until	date		Yes	NULL				
vehicle	varchar(255)		Yes	NULL				
from_addr_id	int(11)		Yes	NULL		-> address.id ON UPDATE RESTRICT ON DELETE RESTRICT		
to_addr_id	int(11)		Yes	NULL		-> address.id ON UPDATE RESTRICT ON DELETE RESTRICT		
driver_id	int(11)		No					

3 edited_order

Creation: Nov 16, 2017 at 09:59 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
driver	varchar(255)		Yes	NULL				
from_name	varchar(255)		Yes	NULL				
from_plz	int(11)		No					
from_street	varchar(255)		Yes	NULL				
from_town	varchar(255)		Yes	NULL				
number_of_animals	int(11)		No					
order_id	int(11)		Yes	NULL				
order_status	varchar(255)		Yes	NULL				
start_time	datetime		No					
status_message	varchar(255)		Yes	NULL				
time_estimation	float		No					
timeframe	varchar(255)		No					
to_name	varchar(255)		No					
to_plz	int(11)		No					
to_street	varchar(255)		No					
to_town	varchar(255)		No					
type_of_animal	varchar(255)		No					
until	date		No					
vehicle	varchar(255)		Yes	NULL				

4 new_order

Creation: Nov 16, 2017 at 09:57 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
driver	varchar(255)		Yes	NULL				
from_name	varchar(255)		Yes	NULL				
from_plz	int(11)		No					
from_street	varchar(255)		Yes	NULL				
from_town	varchar(255)		Yes	NULL				
number_of_animals	int(11)		No					
order_id	int(11)		Yes	NULL				
order_status	varchar(255)		Yes	NULL				
start_time	datetime		No					
status_message	varchar(255)		Yes	NULL				
time_estimation	float		No					
timeframe	varchar(255)		No					
to_name	varchar(255)		No					
to_plz	int(11)		No					
to_street	varchar(255)		No					
to_town	varchar(255)		No					
type_of_animal	varchar(255)		No					
until	date		No					
vehicle	varchar(255)		Yes	NULL				

5 user

Creation: Nov 16, 2017 at 09:57 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
email	varchar(255)		No					
enabled	bit(1)		No					
name	varchar(255)		No					
password	varchar(30)		No					
role	varchar(255)		Yes	NULL				

6 vehicle

Creation: Nov 16, 2017 at 09:57 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
image	varchar(255)		No					
name	varchar(255)		No					
number_of_vehicles	int(11)		No					

4.3 Software System Attributes

4.3.2 Availability

- Version 1
    - There's a working internet connection and the server is up and running.
    - The product shall be available for use between the hours of 8:00am and 5:30pm
    - The product shall be capable of processing up 50 customers.
    - uptime should be around 95% availability

Version 2

- The product shall continue to operate in local mode whenever it loses its link to the central server
- The product shall be capable of processing up to 500 within three years.

4.3.3 Security

- The user data is kept as securely as necessary. Especially the password is stored securely (using the BCryptBCryptPasswordEncoder from the Spring Framework, a hashing algorithm with randomly generated salt).

4.3.4 Maintainability

Version 1

- The product is expected to run under Windows 10 and macOS High Sierra.

Version 2

- The maintenance releases will be offered to end-users once a year.
- Every registered user will have access to a help site via the Internet.
- The product shall be able to be installed in the specified environment within 2 working days.

Out of Scope

- The product might eventually be sold to a foreign market

4.3.5 Portability

- The application should run stable on the browsers Microsoft Edge and Apple Safari 11.0.
- The application should run stable on major OS systems (Windows 10.0, macOS High Sierra).
-

4.3.6 Usability

Version 1

- The product shall help the user to avoid making mistakes
- The product shall make the users want to use it.
- The product shall be used by people with no training
- The product shall be easy for a truck driver to learn.
- The product shall use symbols and words that are naturally understandable by the user community.

Version 2

- The product shall conform to the Swiss Disabilities Act.
- The product shall allow the user to select a chosen language.