



Road Safety Data, Collection, Transfer and Analysis

Deliverable 2.2.

Specification of Data System

Giustiniani,G., Carroccia,R., Robibaro, M. (2012) Specification of Data System, Deliverable 2.2 of the EC FP7 project DaCoTA

Grant agreement No TREN / FP7 / TR / 233659 /"DaCoTA"

Theme: Sustainable Surface Transport: Collaborative project

Project Coordinator:

Professor Pete Thomas, Vehicle Safety Research Centre, ESRI
Loughborough University, Ashby Road, Loughborough, LE11 3TU, UK

Project Start date: 01/01/2010

Duration 30 months

Organisation name of lead contractor for this deliverable:

Vehicle and Traffic Safety Centre, Chalmers University of Technology (SAFER)

Report Author(s):

CTL Centre for Transport and Logistics – Rome – ITALY

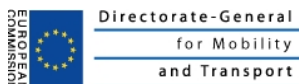
R. Carroccia, M. Robibaro, G. Giustiniani

Due date of deliverable	17/02/2012	Submission date:	17/02/2012
--------------------------------	-------------------	-------------------------	-------------------

Project co-funded by the European Commission within the Seventh Framework Programme

Dissemination Level

PU	Public
-----------	---------------



Project co-financed by the European Commission Directorate General for Mobility and Transport

ACKNOWLEDGEMENTS

The DaCoTA system was developed starting from the System developed by Safer consortium for the Swedish in-depth investigation activities. The Safer consortium made available its system for free. The System was then updated and improved according to the DaCoTA WP2 needs and indications by CTL.

This document describes the DaCoTA system architecture and the opportunity to install a local deployment of the system.

There are also detailed system requirements and installation process.

Descriptions of the main functionalities are present.

It is intended for IT personnel of those partners who want to install a local instance of the information system on their own servers.

TABLE OF CONTENTS

1. introduction	7
2. CORE SYSTEM ARCHITECTURE	8
2.1. Overview of DaCoTA system	8
2.1.1. <i>The three tiered structure</i>	8
2.2. The application: server tier	9
2.2.1. <i>Annotating the business object</i>	9
2.2.2. <i>The business service: database connectivity</i>	9
2.3. The application: client tier	9
2.3.1. <i>MXML</i>	10
2.3.2. <i>ActionScript</i>	10
2.4. The application: storage tier.....	10
3. Functional model: Local servers and central server	11
3.1. Input data	12
3.2. Browsing data.....	13
3.3. Uploading process from LS to CS.....	13
3.3.1. <i>Considerations when deciding between LS or CS Using of Central System</i>	15
3.3.2. <i>Using of Local Server</i>	15
4. DEPLOYMENT OF THE INFORMATION SYSTEM	16
4.1. SYSTEM REQUIREMENTS	16
4.1.1. <i>SERVER REQUIREMENTS</i>	16
4.1.2. <i>BROWSER REQUIREMENTS</i>	17
4.2. INSTALLATION DESCRIPTION	17
4.2.1. <i>TOMCAT</i>	17
4.2.2. <i>INSTALLATION PACKAGES</i>	17
4.2.3. <i>DATABASE</i>	17
4.2.4. <i>INSTALLATION</i>	18
4.2.5. <i>BROWSER</i>	18
4.2.6. <i>SESSION EXPIRY</i>	18
4.2.7. <i>UPDATE ON-LINE MANUAL SHORT DESCRIPTIONS</i>	18
4.2.8. <i>PORTS TO BE OPENED ON THE FIREWALL</i>	18
4.2.9. <i>Repository for files and images uploaded</i>	19
4.2.10. <i>APACHE HTTP/HTTPS WEB SERVER (OPTIONAL)</i>	19
5. Stacks	20
5.1. Protocol Stack	20
5.2. Integration Stack	20

6. Libraries.....	22
6.1. Server.....	22
6.2. Client.....	25
6.3. Development.....	26
6.3.1. <i>Installed during development</i>	26
6.3.2. <i>Used during build</i>	27
7. Main Functionalities.....	28
7.1. System and Case Roles.....	28
7.2. Login.....	31
7.2.1. <i>Scenario: valid login</i>	31
7.2.2. <i>Scenario: Login with a non-valid username</i>	31
7.2.3. <i>Scenario: Login with a non-valid password</i>	31
7.2.4. <i>Scenario: Agree to EULA (End User License Agreement)</i>	32
7.2.5. <i>Scenario: Do not agree to EULA</i>	32
7.2.6. <i>Scenario: Log out</i>	33
7.2.7. <i>Scenario: Log out by session time out</i>	33
7.2.8. <i>Scenario: Log Out by closing windows</i>	33
7.3. User/Case Administration.....	34
7.3.1. <i>Scenario: Accessing the user administration</i>	34
7.3.2. <i>Scenario: Accessing user administration with insufficient permission</i>	34
7.3.3. <i>Scenario: Adding a new Case</i>	35
7.3.4. <i>Scenario: Edit a Case</i>	36
7.3.5. <i>Scenario: Change Case Status</i>	37
7.4. Input Accident data.....	38
7.4.1. <i>Scenario: Input Accident Data</i>	39
7.5. Input Car Data.....	41
7.5.1. <i>Scenario: Create a Car object</i>	41
7.5.2. <i>Scenario: Delete a Car object</i>	42
7.5.3. <i>Scenario: Input Car data – Admin Tab</i>	43
7.6. Input Bus Data.....	44
7.6.1. <i>Scenario: Create a Bus object</i>	44
7.6.2. <i>Scenario: Delete Bus</i>	45
7.6.3. <i>Scenario: Input Bus Data – Admin Tab</i>	45
7.7. Input Truck Data.....	46
7.7.1. <i>Scenario: Create a Truck object</i>	47
7.7.2. <i>Scenario: Delete Truck</i>	48
7.7.3. <i>Scenario: Input Truck Data- Admin Tab</i>	48
7.8. Input Other Vehicle.....	49

7.8.1.	Scenario: Create an Other Vehicle object	49
7.8.2.	Scenario: Delete Other Vehicle	50
7.8.3.	Scenario: Input Other Vehicle Data – General Tab	50
7.9.	Input Pedestrian Data	51
7.9.1.	Scenario: Create a Pedestrian	51
7.9.2.	Scenario: Delete Pedestrian	52
7.10.	Input Bicycle Data	52
7.10.1.	Scenario: Create a Bicycle	52
7.10.2.	Scenario: Delete Bicycle	53
7.11.	Input PTW Data	54
7.11.1.	Scenario: Create a PTW	54
7.11.2.	Scenario: Delete PTW	55
7.12.	Input Road Data	55
7.12.1.	Scenario: Create Road Object	56
7.12.2.	Scenario: Delete Road Object	56
7.12.3.	Scenario: Input Road Data	57
7.13.	Input Road User Data	57
7.13.1.	Scenario: Create a Road User	57
7.13.2.	Scenario: Delete a Road User	59
7.14.	Input DREAM	59
7.14.1.	Scenario: Create a DREAM Analysis	59
7.14.2.	Scenario: Delete a DREAM Analyses	60
7.14.3.	Scenario: Create a Chart	60
7.15.	Input Reconstruction	60
7.15.1.	Scenario: Create Reconstruction Object	60
7.15.2.	Scenario: Delete Reconstruction Object	60
7.16.	Input Injury Analysis Data	61
7.16.1.	Scenario: Create an Injury Analysis	61
7.16.2.	Scenario: Delete an Injury Analyses	62
7.17.	Export	62
7.17.1.	Scenario: Export Data	62
7.17.2.	Scenario: Export Images and Files	63
7.18.	Handle images and upload files	64
7.18.1.	Scenario: Upload image	64
7.18.2.	Scenario: Edit image	65
7.18.3.	Scenario: Delete Image	67
7.18.4.	Scenario: Categorize image	67
7.18.5.	Scenario: (Re)Name Image	67

7.18.6. Scenario: Use in case report	70
7.19. Show and print Case	70
7.19.1. Create Case Report	70
7.19.2. Case Report Layout	71
7.20. Comment.....	72
7.20.1. Scenario: Adding a comment on input fields	72
7.21. Tool tip.....	73
7.21.1. Scenario: Read short description.....	73
7.21.2. Scenario: Go to the wiki.....	73
7.22. Short keys	74
7.22.1. Scenario: Combo Box.....	74
7.22.2. Scenario: Text, Number or Date fields.....	75

1. INTRODUCTION

The main aims of DaCoTA WP2 is to set up a Pan-European In-depth Accident Investigation Network and to provide the investigation teams of the network with common methodology and tools for carrying out in-depth investigations at a national level.

One of the tools that should be provided to the teams of the network is a Database in order to store, analyse and exchange the in-depth data collected by the different teams.

This Database, called the DaCoTA System has been developed within the framework of Task 2.4 starting from the System developed by Safer consortium for the Swedish in-depth investigation activities. The Safer consortium made available its system for free. The System was then updated and improved by CTL according to the DaCoTA WP2 needs and indications.

This document describes the DaCoTA system architecture and the opportunity to install a local deployment of the system. There are also detailed system requirements and installation process. Descriptions of the main functionalities are present.

It is intended for IT personnel of those partners who want to install a local instance of the information system on their own servers.

2. CORE SYSTEM ARCHITECTURE

2.1. Overview of DaCoTA system

In Figure 2-1 is reported an overview of the DaCoTA system architecture. The Figure is a fairly detailed outline that shows the complexity of the system. It is not a simple database or a simple application. It is a Rich Internet Application (RIA) and needs the use of the right set of technologies to provide the service.

Adobe Flex is a client-side technology that provides developers with a rich set of API calls for creating GUIs (Graphical User Interface), drawing graphics, playing and streaming media, and connecting to Web services. On the server side, Java technology provides abilities such as connecting to relational database management systems (RDBMSs), multi-threaded processing of service requests, and optimum scaling with increased demand. The joint use of these two technologies offers a powerful technology stack that satisfies the demand of RIA applications.

The system utilizes Flex for the client, Java technology for the server, and PostgreSQL for the back-end database.

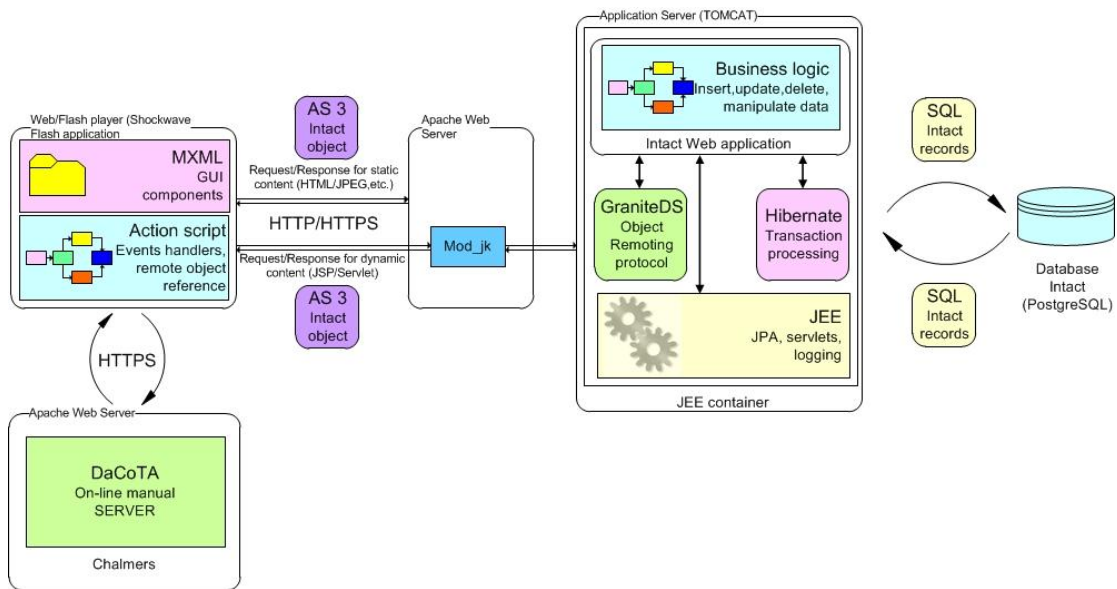


Figure 2-1 - The DaCoTA system architecture

These choices for the architecture and the technologies have been made by the team that started to develop the information system originally for the Swedish “Intact” project.

2.1.1. The three tiered structure

The DaCoTA system provides a rich UI that supports the insertion, modification and deletion of accident data (text, images, documents) through an Adobe Flash (SWF) application. This three-tiered Web architecture is depicted in the above figure, where the client is represented by the SWF file embedded within a Web page. The server application is run within a Java servlet container (in this case, Apache Tomcat) and the database is PostgreSQL. Combined, these three tiers create a power-distributed application and the application potentially can be divided.

In addition, the application connects to an on-line manual (so called Wiki) system showing a popup with information about the selected topic in the tooltip associated with the variable selected displayed on the web form.

For communication between the Flash application and the Java Servlet container, the GraniteDS framework provides object remoting - a form of RPC that allows Adobe ActionScript objects to call Java objects and vice versa. Communication between the Java server application and the relational database is handled by the Hibernate Object Relational Mapping (ORM) framework. Hibernate allows Java objects to be changed to SQL code and vice versa.

Applications are usually broken into logical chunks called "tiers", where every tier has an assigned role. Traditional applications are composed only by 1 tier (e.g. SafetyNet System), which resides on the client machine, but web applications need a n-tiered approach by nature. Many variations are possible but the most common structure is the three-tiered application. In its most common form, the three tiers are called *presentation*, *application* and *storage*, and it is what is implemented in the DaCoTA system. A web browser with Adobe Flash (SWF) application is the first tier (presentation), an engine using some dynamic Web content technology (JSP/Java) is the middle tier (application logic), and a database is the third tier (storage). The web browser sends requests to the middle tier, which services them by making queries and updates against the database and generates a user interface.

2.2. The application: server tier

A set of Java classes that encompass the information required to store, retrieve and manipulate accident's information have been developed.

2.2.1. Annotating the business object

The Java classes are considered POJO (Plain Old Java Object) that act as business objects, meaning that they represent business domain characteristics and behaviors. The data inside DaCoTA objects needs to be persisted to the database. The solution is to use an ORM framework such as Hibernate, which performs much of the work in mapping objects to records within database tables and back again.

Each time an in-memory Java Object is required to be persisted, Hibernate transforms the state information of any Java objects into an SQL update. Likewise, SQL statements with result sets are used to populate Java objects. The result is that all objects can be saved as records within the database, and all records can be retrieved and transformed back into Java objects.

2.2.2. The business service: database connectivity

A service class is required to perform the calls to Hibernate in order to execute ORM.

2.3. The application: client tier

The Flex framework allows you to create applications that users can play in Adobe Flash Player. Flex consists of:

- A declarative XML UI language known as MXML;
- The ActionScript programming language;
- Run time libraries for creating UIs, Web connectivity, and many other features;

- Developer tools for compiling applications into SWF files.

The client application referenced in DaCoTA uses Flex version 3. Before approaching the client-side application, it is important to understand how Flex applications are created and how they exist as executables within Flash Player.

First, it is possible to create applications using a combination of MXML markup and ActionScript code. A common workflow is to create much of the GUI (presentation) using the MXML format, and then use ActionScript code for event handling and business logic. Because both MXML and ActionScript are text-based, a standard text editor and the Flex SDK are all you need to create Flash applications.

Second, once a Flex application has been written, the code must be compiled using the MXML compiler. The MXML compiler creates SWF files that can then be run inside a Web browser (via the Flash Player browser plug-in).

Finally, Flash applications run within the ActionScript Virtual Machine 2 (AVM2).

2.3.1. MXML

MXML is a powerful declarative XML format that helps to:

- Minimize the amount of code required to build a GUI because of the declarative nature of the XML format;
- Reduce the complexity of GUI code by allowing for a clear separation of presentation logic and interaction logic;
- Promote use of design patterns when approaching software development.

2.3.2. ActionScript

While MXML defines the GUI, ActionScript offers the behavior for handling events, binding data (through the Bindable metadata tag), and the ability to call a remote service.

These ActionScript objects are sent to the server side, where GraniteDS performs its activity and converts the ActionScript objects into Java objects.

2.4. The application: storage tier

The storage tier consists of the RDBMS PostgreSQL and implements the data persistency. The database consists of about 100 tables and many relationships. The database has also a duplicated instance where is saved the history of all modifications in the data tables.

3. FUNCTIONAL MODEL: LOCAL SERVERS AND CENTRAL SERVER

The Central information System (Web application and database), hereinafter CS, will be located in Rome at CTL office. Some partners asked for the possibility to have a Local System (LS hereafter) where to insert their data and review them before sharing with the partnership using the CS. In the following will be described how the original monolithic DaCoTA system has been split into CS and LSs.

We have defined two separate types of Systems that can communicate using a safe connection (SFTP). Some tables in the database contain data to characterize each of the two systems and also specific features of the web applications are provided to manage the two different situations.

Each user of each partner that has both access to CS and to LS will be initially provided with two different sets of usernames and passwords; one to login to the CS and the other one to login to LS.

Each partner will be able to input accidents data on its own local system (LS) or to input them directly on the central system (CS). The choice is exclusive: it is not allowed to input data on both LS and CS for the same partner. The research center must choose between LS and CS once and for all, before starting the input of cases. So there is no possibility of conflicts between entries.

The proposed architecture is shown in the following Figure 3-1.

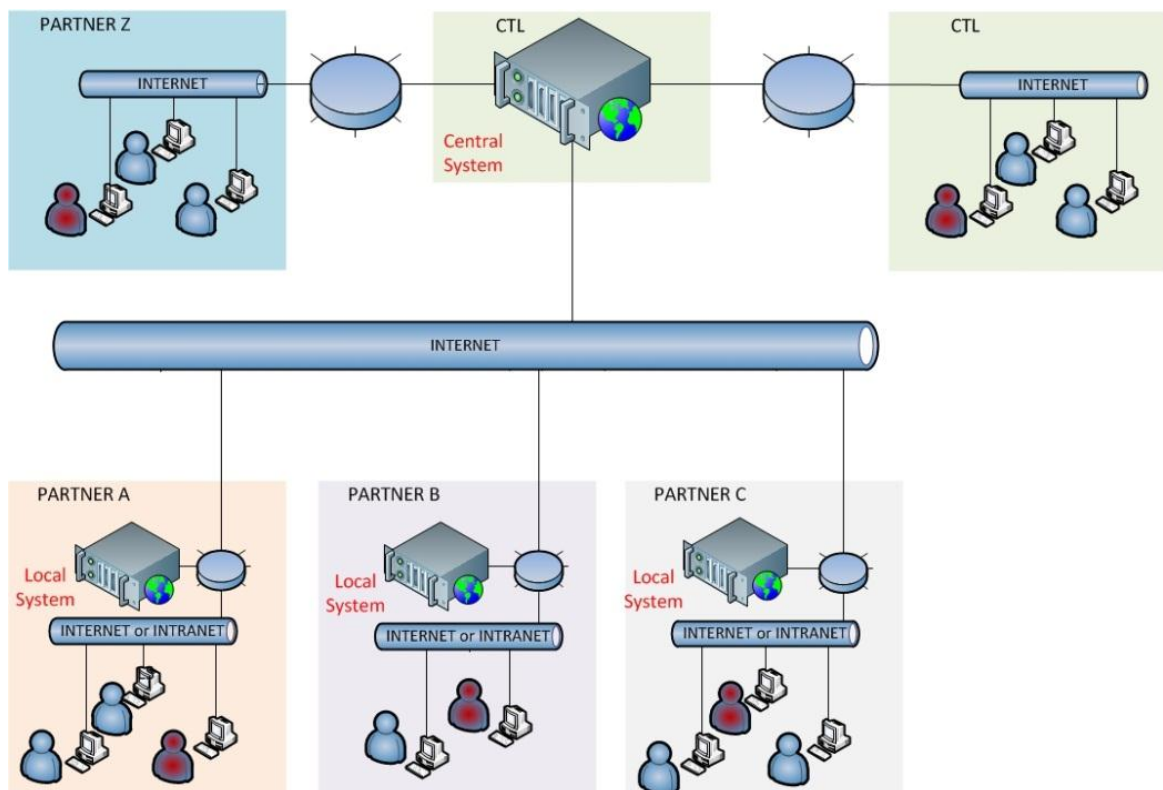


Figure 3-1 - Architecture with Central System and Local Systems.

3.1. Input data

Each partner will be able to input accidents data on its own local system (LS) or to input them directly on the central system (CS). The choice is exclusive: it is not allowed to input data on both LS and CS for the same partner.

- LS input data choice: Authorized users of partner who wants to enter data on their own LS (with administrative rights), have a command in the client web interface menu to manage uploading of accidents data to CS. An alert is available informing the user when completed and published accident cases are not yet uploaded. By clicking on a specific command button, a new web window will be opened and the user can choose, among the published accidents cases, which to transfer into the CS using a secure protocol (sftp). It is important to remark that only authorized users for each partner will have the rights to upload data to the CS.
- CS input data choice: Users of partners who decided to input accident data directly onto the CS, will have direct access to it. The accident data will be stored and will remain only on the CS database during their life cycle.

The user can browse the LS and CS with an internet browser separately opened in different windows.

A view of the input process is reported in Figure 3-2.

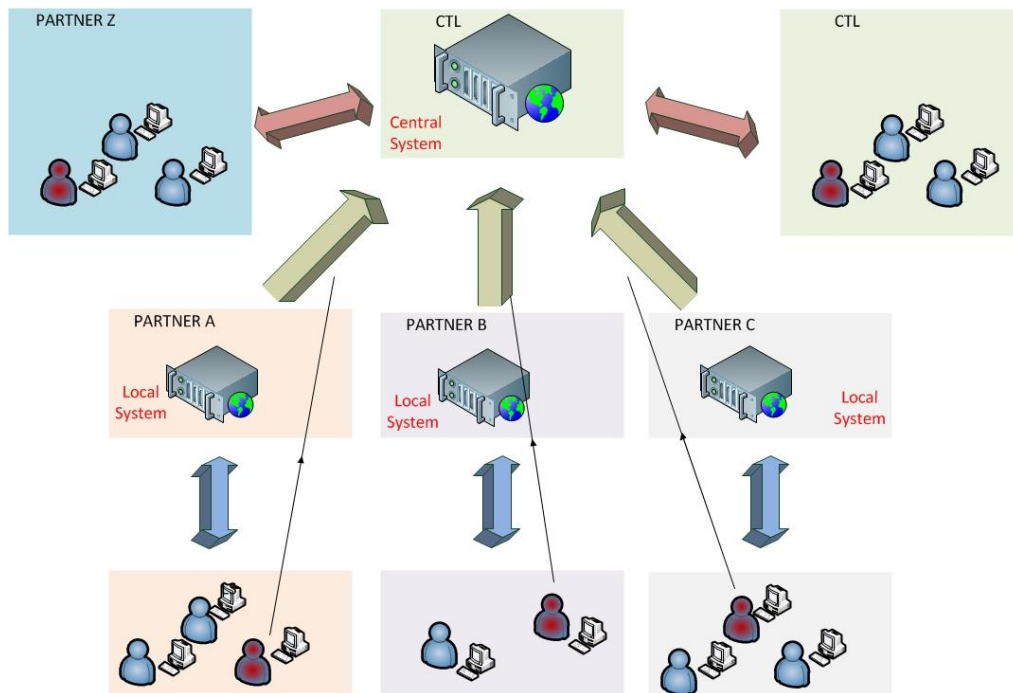


Figure 3-2 - Input data.

Partners that have a LS can input data only on their systems, but they can browse the accident cases on CS as well as on the LS and download the raw data.

Users that don't have a LS, can input data on the CS and browse and download data on the CS.

3.2. Browsing data

Any partner provided with a LS can allow other partners to browse its LS by providing them with password and username. An overview of all these functionalities is reported in Figure 3-3.

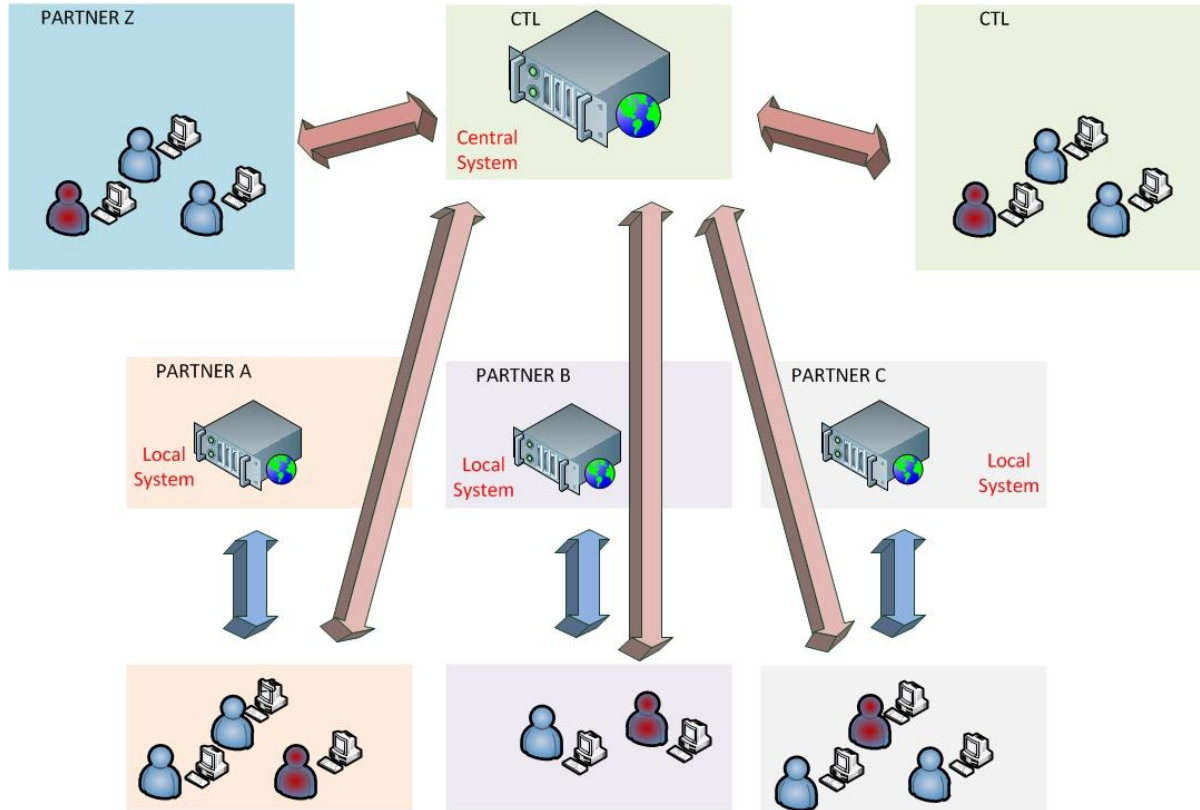


Figure 3-3 - Browsing data.

CS and LS will be provided with a login control system that will allow to establish a connection to the web applications only from authorized IP addresses (a white list of possible IP numbers, or some allowed interval of IP numbers must be available for each partner). This rule should deter unauthorized access to the system from non-permitted workstations. Connections to the systems will be possible only from trusted locations with known static IP addresses.

3.3. Uploading process from LS to CS

A scheduled daemon process running on the CS, once a day (during night hours), will get the compressed files (zip archive) transferred during the day from different LSs, decompress them obtaining CSV (Coma Separated Values) files and copy data to the corresponding tables of PostgreSQL database on the CS. This process is reported in Figure 3-4.

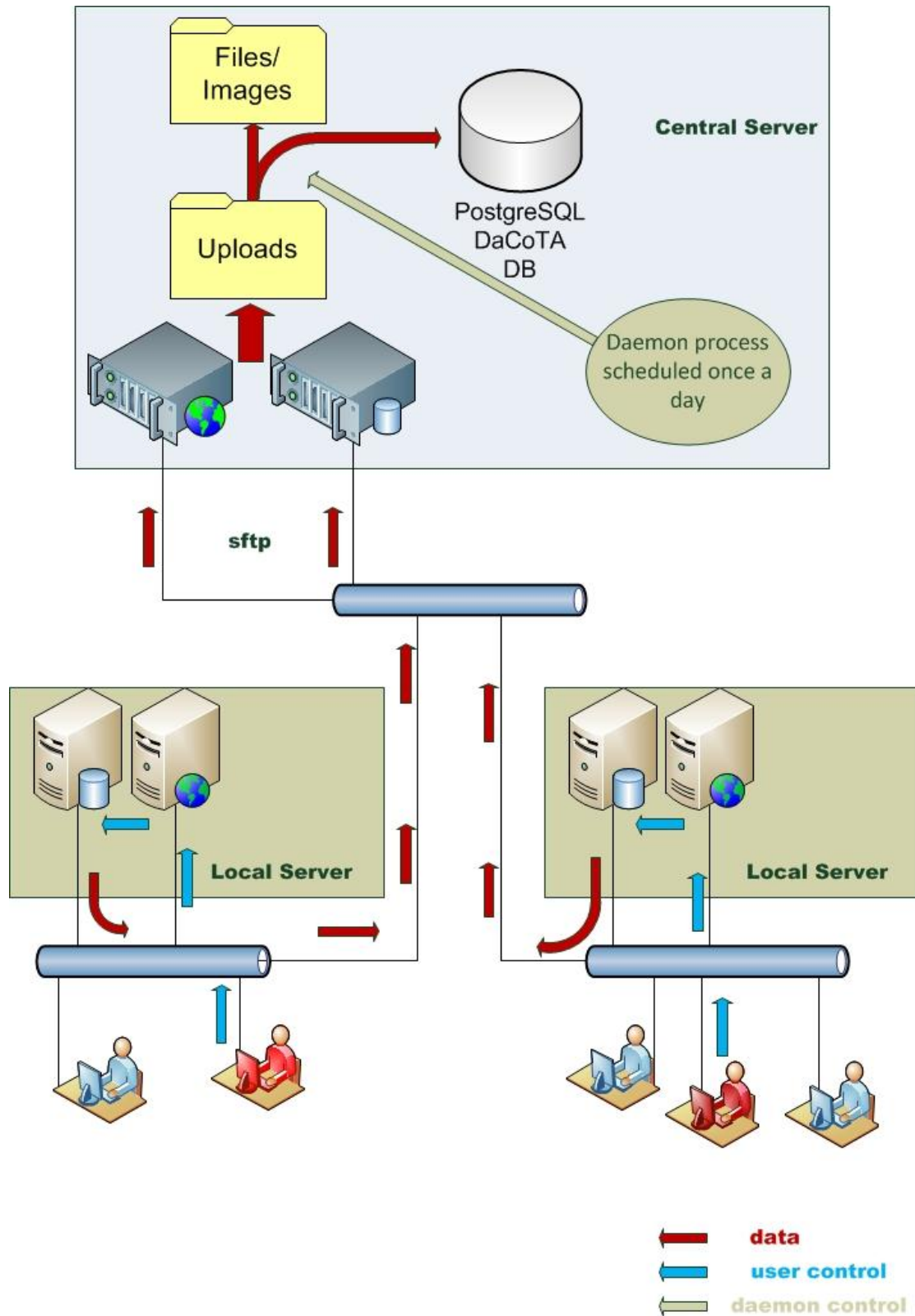


Figure 3-4 - Details of uploading process from a LS and data load on CS

Only authorized users with administrative rights for each partner LS can upload data on the CS. Only completed and published accident cases will be uploaded. If a partner would like to share data of its accidents not yet completed with other partners, they will do so by allowing the outside users access to its LS.

3.3.1. Considerations when deciding between LS or CS Using of Central System

If a partner decides to use the CS instead of LS:

- The servers and all the hardware accessories (UPS, network routers, network switches, firewall, storage devices, etc.) will be provided by CTL. Thus, partners will not have additional support costs (resulting in a cost savings for the partner).
- During the DaCoTA project, CTL will be the Administrator of the system and will provide all the software and management assistance to the system. Additionally, all the security issues related to ensuring data integrity and confidentiality will be managed by CTL. Partners will save time and cost of their IT personnel.
- Accident data of each partner will be loaded and stored physically on the CTL servers in Italy. Partners will not have their data stored in a local database. Each partner can download its data in CSV (Comma Separated Values) format – directly available to be opened in Microsoft Office Excel – available at any time. Each partner remains the sole owner of their data.
- At the end of the project, the CS will no longer be maintained by CTL.

3.3.2. Using of Local Server

If a partner decides to use LS instead of the CS:

- Each partner must have the hardware (servers, UPS, network accessories) and the software installed and configured to host the application (resulting in additional costs)
- Each partner needs to have skilled IT personnel to administrate and manage the LS. Each time a new release of the application or a modification on the DB structure will be implemented by CTL in consequence of the project development guidelines, the modifications have to be transferred to the local servers by skilled personnel. Typically, they have to copy dacota.war file in the appropriate folder and run sql scripts inside PostgreSQL RDBMS.
- IT personnel will be provided with a guide for the deployment of the DaCoTA software platform.
- Partner data will be loaded and stored on local server in a database management system. Only a copy will be sent to the CS in Italy and stored in another database that collects data coming from all partners.

4. DEPLOYMENT OF THE INFORMATION SYSTEM

4.1. SYSTEM REQUIREMENTS

The following server and client requirements must be met before deploying the DaCoTA system.

4.1.1. SERVER REQUIREMENTS

The DaCoTA server can be deployed in any environment (e.g. Unix-like systems, Linux or Windows operational system), as long as the following software is available:

- Apache Tomcat/6.0.20 (<http://tomcat.apache.org/download-60.cgi>);
- PostgreSQL 8.4.1 or successive versions (<http://www.postgresql.org/download/>).

The hardware requirements are:

- 4GB RAM with 2GB reserved for Tomcat.
- The storage capability of the server must be fixed according to the amount of data that is supposed to be collect. This amount of data must take in account the number and the size of the photos, videos and other documents that the users are going to upload to the servers. A storage capability of 250GB for the data should be enough.

A server architecture is highly recommended to guarantee good performances: hardware redundancy for uninterruptable power supplies and hard disks arranged so that if one fails, another is automatically available, good internet access, increased security, advanced backup facilities to permit regular and frequent backups of critical data, and so on. These are common requirements of server architectures:

- Redundant power supply.
- UPS (Uninterruptible Power Supply) that provides emergency power to the servers when the input power source, typically mains power, fails.
- Storage data distributed across the server hard disk drivers at least in RAID 1 (better if in RAID 5 or greater).
- Bandwidth to access to the server is limited by the worst bandwidth of the chain of connections between the browsing computer and the remote server. Although, if the server is interfaced with the external internet network with a bandwidth of 100Mbit/sec or 1Gbit/sec, it is a good rate to avoid traffic latency due to simultaneous access to the server itself.
- To increase the security of your system, it is a good practice to:
 - Use two different hardware machines to host the web server (web application) and the data server (DBMS PostgreSQL);
 - Use a firewall to filter out-going and in-coming connections to the web server;
 - Define a white list of IP addresses that can connect to the web server and deny access to unknown IPs.
- Scheduling of database data backups done at least once a day, together with the files and images associated to the accidents cases and present in some systems folders of the file systems (in Unix-like systems, the folder

is `/var/tmp/dacota/repo`, in Windows systems the folder is `C:\tmp\dacota\repo`. The full content of the repository folder must be backed up in sync with the database to keep data integrity.

4.1.2. BROWSER REQUIREMENTS

The DaCoTA client is platform independent, which means that it will run on any OS with a browser that supports Adobe Flash.

- Adobe Flash Player 10 (<http://get.adobe.com/flashplayer/>).

Adobe Flash Player 10 is a minimum version requirement. The application has primarily been tested in Windows XP and Vista with Firefox 3 and Internet Explorer 7 browsers.

The viewing experience of the DaCoTA Client is tested and optimized for a screen resolution of at least 1280x1024.

4.2. INSTALLATION DESCRIPTION

The following text describes the installation process steps. A prerequisite for this guide is that the software listed in the system requirements section has been installed.

4.2.1. TOMCAT

Tomcat's available memory must be increased. This is done by setting the environment property `CATALINA_OPTS` each time before Tomcat is started.

- `CATALINA_OPTS="-Xmx2048m -XX:MaxPermSize=1024m"`.

This can be written in `.bashrc` for the user `tomcat` (or whichever user the tomcat process runs as).

Open `/home/Tomcat/.bashrc` and add:

- `export CATALINA_OPTS="-Xmx2048m -XX:MaxPermSize=1024m"`

4.2.2. INSTALLATION PACKAGES

The package produced for installation and upgrade contains two files.

- `dacota.war`;
- `database_upgrade_2011xxxx.sql`.

The `dacota.war` file is in Web application ARchive format which allows it to be unpacked and installed on a Tomcat server in a standardized way. The `.sql` file included is a series of sql commands that must be run on the PostgreSQL database.

4.2.3. DATABASE

1. If installing for the first time, a database on the same server that Tomcat is run has to be created:
 - a. Name: intact;
 - b. User: intact, with all rights in the database;
 - c. Password: intact.

2. The file `database_upgrade_2011xxxx.sql` needs to be executed on the database intact. (as alternative, we could produce a backup file for the database to be restored on your PostgreSQL DBMS).

4.2.4. INSTALLATION

1. Stop Tomcat;
2. Remove `dacota/` and `dacota.war`, if they are in the `webapps` directory;
3. This file `dacota.war` should be placed in the Tomcat installation directly in the `webapps` directory;
4. Start Tomcat.

4.2.5. BROWSER

The user browses the url `http://example.com/dacota`.

4.2.6. SESSION EXPIRY

If the user is idle for a long period of time, it is logged out from the system automatically. This is referred to as the session expire time. The length of this period is determined by the container that the application is deployed in. For example, in Tomcat the time can be set in the manager application.

Documentation for the Tomcat manager application can be found at: <http://tomcat.apache.org/tomcat-6.0-doc/manager-howto.html>.

4.2.7. UPDATE ON-LINE MANUAL SHORT DESCRIPTIONS

To update the Wiki short description values and to align the content of the online manual to the content on the database, it's necessary to run a Python script, scheduled on your operating system to run once a day (e.g. during the night-hours) that automatically will execute this task.

For this task, it is required to have installed:

- Python 2.7.2
- Psycopg2 - 2.4.2

4.2.8. PORTS TO BE OPENED ON THE FIREWALL

For some operations between the LS and the CS and between the DaCoTA system and the online manual portal, some communicating ports must be opened on your firewall:

- Port 22: SFTP (Secure File Transfer Protocol) with username and password. On this channel, the following should be performed:
 - Activation of the local web applications;
 - Uploading data of accident cases from LSs to the CS.
- Port 443: HTTPS (Hyper Text Transfer Protocol Secure) with username and password. On this channel, the following should be performed:
 - Updating the online manual short descriptions.

4.2.9. Repository for files and images uploaded

- The files and images uploaded for the accident cases are stored in the file system reserved folders:
 - Unix-like systems: /var/tmp/dacota/repo
 - Windows systems: C:\tmp\dacota\repo

4.2.10. APACHE HTTP/HTTPS WEB SERVER (OPTIONAL)

Apache Tomcat is a Servlet/JSP container and used to deploy dynamic Java contents like JSPs and Servlets. Tomcat has some web server capabilities, however it is not a full web server designed to serve high traffic web sites. Apache HTTP server is instead a full web server. So in general Tomcat is configured along with Apache HTTP server to achieve a web site with dynamic content well-designed and well-balanced. (<http://httpd.apache.org/>).

5. STACKS

5.1. Protocol Stack

Most requests from the Flash-application are handled using Tide. Tide is a framework for allowing Spring service bean methods to be called from the client side with very little effort. The type of calls made include loading case data, updating case data, and login.

The AMF3 protocol is a binary format that includes optimizations for communication with a Flash-application.

Files are uploaded using a regular HTTP file upload. The *multipart/form-data* mime encoding is used to allow files larger than 2 megabytes. Downloads of images and files from the webserver use the regular HTTP protocol.

Downloaded files are the intact.swf file which is the Flash-application, images and various types of documents that belong to a case, export files (csv), and report files (pdf).

Uploaded files are images and documents that belong to a case. An overview of protocol stack is reported in Figure 5-1:

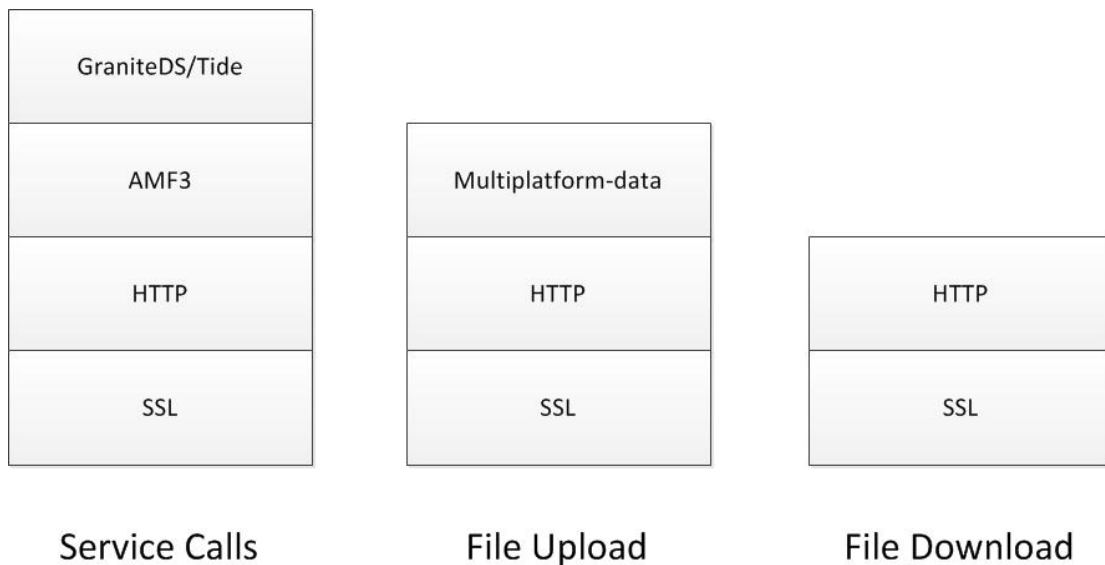


Figure 5-1 Protocol stack

5.2. Integration Stack

An overview of the integration stack is reported in Figure 5-2. The main features of integration stack are:

- The core of the server is wired together using the Spring Framework.
- Hibernate is used to make calls to the database.
- The GraniteDS Tide Framework has components on both server and client side to handle communication between the DaCoTA client application and services wired by Spring.

- Flash has built-in support for file upload and download. On the server side, Jersey handles all HTTP calls by the Flash application.

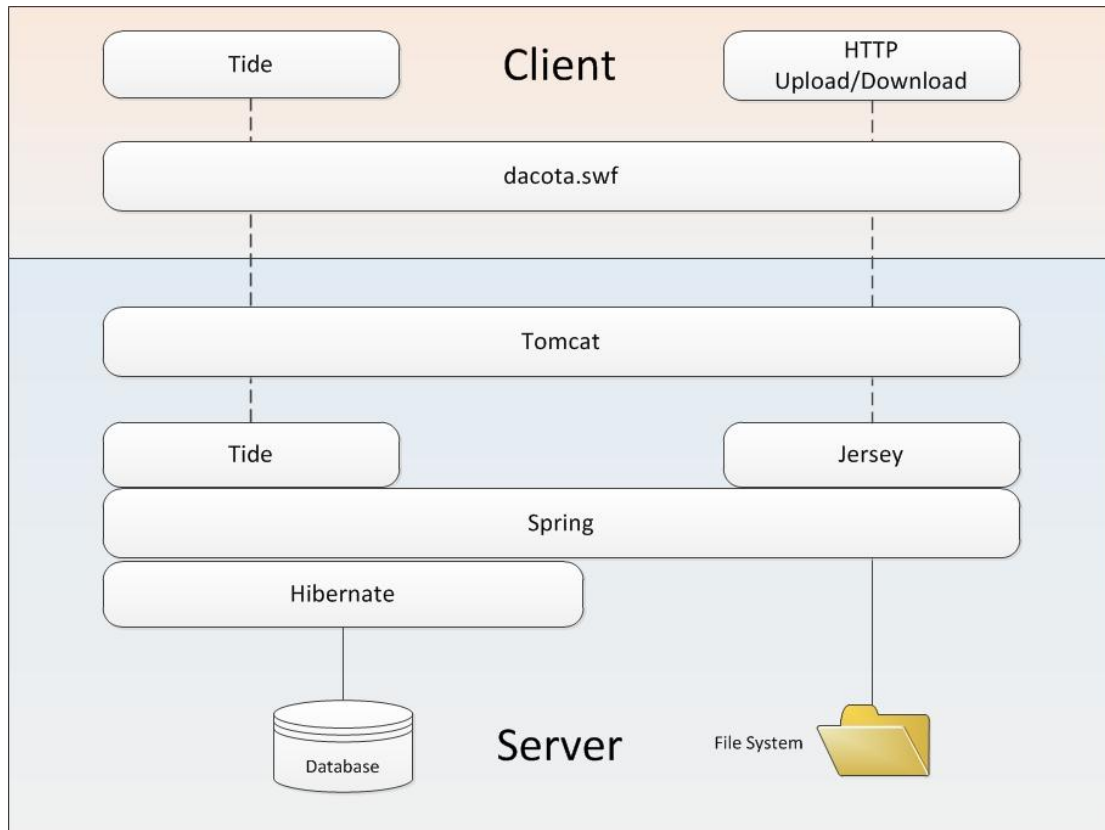


Figure 5-2 Integration stack

6. LIBRARIES

Libraries in the DaCoTA project are fetched using Maven during the build process and not distributed with the source code, with the exception of BIRT.

6.1. Server

The following libraries are distributed with the server.

Software	Versions	Link	Comments
Spring Framework	Version 2.5.6; Apache License, Version 2.0	http://www.springsource.org/about	The Spring Framework is an application framework. The framework is used in DaCoTA for dependency injection and separation of concerns.
Spring Security	Version 2.0.4; Apache License, Version 2.0		Handles authentication and authorization of users
GraniteDS	Version 2.0.0.SP1-REV3075; GNU Lesser General Public License, version 3	http://www.graniteds.org/	Granite Data Services is an open source alternative to Adobes LiveCycle Data Services. It handles serialization and de-serialization of AMF3. Its sub-module Tide creates necessary code for calling services defined by the Spring Framework. The GAS3 sub-module is a tool for creating ActionScript versions of Java objects that are sent between the server and client so that both ends are strongly typed.
Hibernate	Version 3.5.0-CR-2; GNU Lesser General Public License, version 2.1	http://www.hibernate.org	Hibernate is primarily used as an Object/Relational Mapping tool. In other words it stores and retrieves data from the database with the help of the PostgreSQL JDBC Driver described below.
PostgreSQL JDBC Driver	Version 8.3-603.jdbc3; BSD License Version 9.0-801.jdbc4; BSD License	http://jdbc.postgresql.org	A JDBC driver handles the connection to a database. This is a specialized driver for PostgreSQL.
AspectJ	Version 1.5.3; Eclipse Public License Version 1.6.5; Eclipse Public License	http://www.eclipse.org/aspectj/	An aspect oriented programming library
SLF4J	Version 1.5.2; MIT License	http://www.slf4j.org/	SLF4J is a logging framework that only handles collecting log messages from the code. Another framework is needed to

			write the log-messages somewhere.
Log4J	Version 1.2.14; Apache License, Version 2.0	http://logging.apache.org/log4j/index.html	Log4J is a logging framework for java. In DaCoTA it handles writing of log files on disk.
Jersey	Version 1.4; COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1	http://jersey.java.net/	Jersey is a reference implementation of JSR 311, which is a java interface for creating RESTful services. It is used to handle HTTP Requests (as opposed to AMF3) to the server from the Flash Application. This involves file upload and download including serving images.
BIRT Project	Version 2.5.1; Eclipse Public License Version 2.5.2; Eclipse Public License	http://www.eclipse.org/birt/phoenix/	The Business Intelligence and Reporting Tools (BIRT) Project is used in DaCoTA to create report files of cases in PDF. BIRT is distributed with the source code for DaCoTA.
Apache Commons	Apache License, Version 2.0	http://commons.apache.org/	Apache Commons is a set of libs with solutions to commonly encountered problems in software development.
CGLIB	Version 2.2; Apache License, Version 2.0		CGLIB is used to create bytecode dynamically by various other libraries

6.2. Client

The following libraries are distributed with the Flex application. Thus, they are distributed to every web browser accessing a deployed server. Since the Flex application is distributed with the server, these libraries are distributed with the server as well.

Software	Versions	Link	Comments
GraniteDS	Version 2.0.0.SP1-REV3075; GNU Lesser General Public License, version 3	http://www.graniteds.org/	Granite Data Services is an open source alternative to Adobes LiveCycle Data Services. It handles serialization and de-serialization of AMF3. Its sub-module Tide creates necessary code for calling services defined by the Spring Framework. The GAS3 sub-module is a tool for creating ActionScript versions of Java objects that are sent between the server and client so that both ends are strongly typed.
PureMVC	Version 2.0.4; Creative Commons 3.0 Attribution US License	http://puremvc.org/	PureMVC is a Model View Controller framework for ActionScript. It has been successively phased out of the system and is now only used to create a service registry.
DateUtils	Version 1.0.8; MIT License		Used to simplify date handling in ActionScript.
Process Map	Unlicensed	http://www.degrafa.org/samples/data-visualization.html	A tool used to implement the graphical parts of DREAM.
Degrafa	Version Beta 3.1; MIT License	http://www.degrafa.org/	Used by Process Map.

6.3. Development

Tools used for development but not included in build.

6.3.1. Installed during development

The following libraries are necessary while developing and are installed separately. Source code and scripts that use the libraries are a part of version control, but not included in the distribution.

Software	Versions	Link	Comments
Apache Maven	Version 2.2.1; Apache License, Version 2.0	http://maven.apache.org/	Maven is a build and project management tool. The structure of the DaCoTA code base is based on the conventions decided by the Maven project and its notion of modules.
Ant	Version 1.8.1; Apache License, Version 2.0	http://ant.apache.org/	Ant is a build tool that is used in some areas to assist maven.
MyBatis 3 Migrations	Version 3.0.3; Apache License, Version 2.0	http://www.mybatis.org/java.html	MyBatis Migrations is a system for handling database deltas. It is used to make synchronization of changes to developer databases easier. It can also produce an SQL script to be run on a production database in a structured fashion.

6.3.2. Used during build

The following libraries are used to build the distribution from source code, but are not distributed with the final product.

Software	Versions	Link	Comments
JUnit	Version 4.4; Common Public License - v 1.0	http://www.junit.org	JUnit is used for testing Java code.
Flex Mojos	Version 3.5.0; Apache License, Version 2.0	http://flexmojos.sonatype.org	Flex Mojos allows maven to compile ActionScript into an .swf or .swc file.
Flex SDK	Version 3.5.0.12683, Mozilla Public License Version 1.1	http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK	The Flex SDK contains the mxml binary used by Flex Mojos to compile ActionScript.

7. MAIN FUNCTIONALITIES

In this chapter we will focus our attention on how the “system roles” and “case roles” are organized to grant the users access to the system and how it is possible to use the main functionalities of the information system.

Access to the information system is protected by username and password. Each authorized user has received their own username and password. Using a web browser (e.g. Internet Explorer, Mozilla Firefox, Google Chrome, etc.) with Adobe Flash Player 10 plugin installed, it is possible to connect to the following url:

- <https://www.ctl.uniroma1.it/dacota>

The communication channel is encrypted and secured using https protocol. The server SSL (Secure Sockets Layer) certificate is regularly registered by Actalis Server Authentication Certification Authority and ensures a high-efficiency and strong 256-bit encrypted connection (SHA-1 with RSA encryption).



Figure 7-1 DaCoTA Application Login Page

7.1. System and Case Roles

When Administrators of the system creates a new user, they assign a username to and insert a valid e-mail address. The system automatically generates a random password and sends an email to the new user with the username and password. The first time the user login to the DaCoTA system, the user will be required to change their password.

When the Administrator of the system creates a new users, he must also assign a “system role” to the new member. The available roles for the system are:

- Administrator;
- Operational Manager;
- Team Organizer;
- Investigator;
- Internal Analyst;

- External Analyst.

In addition to the “system role”, each user can have assigned a “case role” to be able to manage the loaded accident cases in the database.

The available roles for accident cases are:

- Member;
- Leader;
- Medical Investigator.

Figure 7-2 summarizes the System Authorities / System Roles and Case Authorities / Case Roles table schema.

	System Administrator	Operational Manager	Team Organizer	Investigator	Internal Analyst	External Analyst	Case Roles	Member	Leader	Medical Investigator
System Authorities										
View/edit user admin	x	x								
Create new case	x	x	x	x						
Case Authorities										
View case admin	x	x	x	x	x			x	x	x
Edit case admin	x	x	x						x	
View in work progress case data	x	x	x	x				x	x	x
Edit work in progress case data								x	x	x
View published case data	x	x	x	x	x	x		x	x	x
Edit published case data										
View finished case data	x	x	x	x	x	x		x	x	x
Edit finished case data										
<i>Note: To edit published or finished case, status must be changed back to "work in progress".</i>										

Figure 7-2 System Authorities / System Roles and Case Authorities / Case Roles table

7.2. Login

7.2.1. Scenario: valid login

A given user is not logged in and has the correct credentials to access the system.

- User inputs a valid username (case sensitive).
- User inputs a valid password (case sensitive).

System does not reveal the password, but “masks it with stars”.

- User presses “enter” key to login.
- Or user “clicks” the login button.

System displays the user agreement.

- User reads the agreement.
- Clicks the check box to accept the agreement

7.2.2. Scenario: Login with a non-valid username

Given user is not logged in and is not registered or is blocked from the system.

- User inputs a username that doesn’t exist in the system.
- User inputs a password.
- User presses “enter” key to login.
- Or user “clicks” the login button.

System displays the message: “Login failed: Bad credentials.” and clears input fields.
Given user is not logged in and is blocked from the system.

- User inputs a username that is blocked.
- User inputs a password.
- User presses “enter” key to login.
- Or user “clicks” the login button.

System displays a message: “Your account has been blocked. Please contact the system administrator for more information.” and clears input fields.
Given user is not logged in and their account has expired.

- User inputs a username whose account has expired.
- User inputs a password.
- User presses “enter” key to login.
- Or user “clicks” the login button.

System displays a message: “Your account has expired. Please contact the system administrator for more information.” and clears input fields.

7.2.3. Scenario: Login with a non-valid password

Given user is not logged in and has a non-valid password.

- User inputs a valid username.
- User inputs a non-valid password.

System displays a message: “Login failed: Bad credentials” and clears input fields.

- User inputs the wrong password for the third time.

The system blocks the user and system displays a message: “You have entered the wrong password too many times. Your account has been blocked by the system.”

7.2.4. Scenario: Agree to EULA (End User License Agreement)

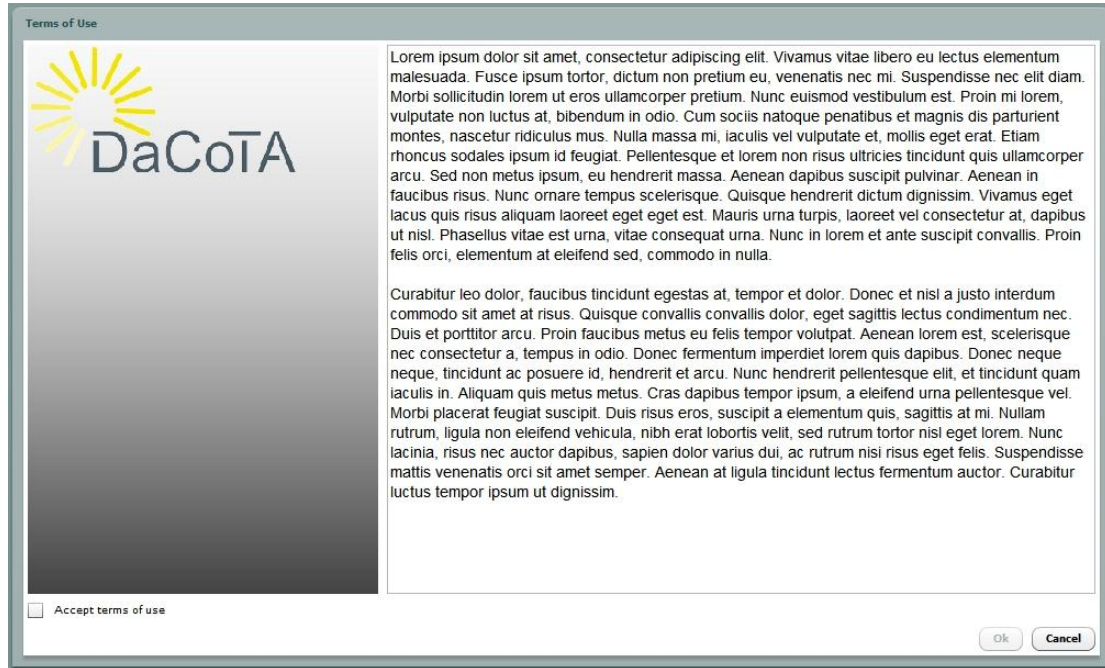


Figure 7-3 EULA page

Given user has entered a valid username and password.

System displays a window with the EULA-text.

- User checks the check box.

System enables the “I accept” button.

- User tabs to the “OK”-button and presses “enter” to accept EULA.
- Or user “clicks” the “OK”-button.

System presents the welcome screen which contains;

- the menu bar
- the information bar
- the case tree
- Alternative A1 – No Cases exist
- System displays a welcome text in the “Main View”.
- Alternative A2 – At least one Case exists
- System displays the “Admin”-tab on the “Accident”-view on the first Case.

7.2.5. Scenario: Do not agree to EULA

Given user has entered a valid username and password.

System displays a window with the EULA-text.

- Alternative A1:

- User checks the check box.
- System enables the “OK”-button.
- User tabs to the “Cancel”-button and presses “enter” to not accept EULA.
- Or user “clicks” the “Cancel”-button.

System closes the “EULA” window and displays the log in view with clean input fields.

- Alternative A2:
 - User unchecks the checkbox.
 - System disables “OK”-button
 - User tabs to the “Cancel”-button and presses “enter” to not accept EULA.
 - Or user “clicks” the “Cancel”-button.

System closes the “EULA” window and displays the log in view with clean input fields.

- Alternative A3
 - User tabs to the “Cancel”-button and presses “enter” to not accept EULA.
 - Or user “clicks” the “Cancel”-button.

System closes the “EULA” window and displays the log in view with clean input fields.

7.2.6. Scenario: Log out

Given user is logged in to the system with valid credentials.

- User clicks the “Logout” link in the Main Menu.

System logs out the user and displays the log in view with clean input fields.

7.2.7. Scenario: Log out by session time out

Given user is logged in to the system with valid credentials.

- User performs no action within a system defined time.

Server drops the client session. When the user performs the next action, a message is displayed and the user is logged out.

- User confirms that they have read the message by clicking OK.
- System displays the log in view with clean input fields.

7.2.8. Scenario: Log Out by closing windows

Given user is logged in to the system with valid credentials.

- User closes all widows.

System logs out the user.

- User opens browser and navigates to the DaCoTA site.

System displays the log In view.

7.3. User/Case Administration

The Administrator, using the case administration menu (see Figure 7-4) can:

- Add/edit Users;
- Add a new case;
- Edit a case;
- Change the password for a given user.

In the following, a short overview of these operations will be reported.

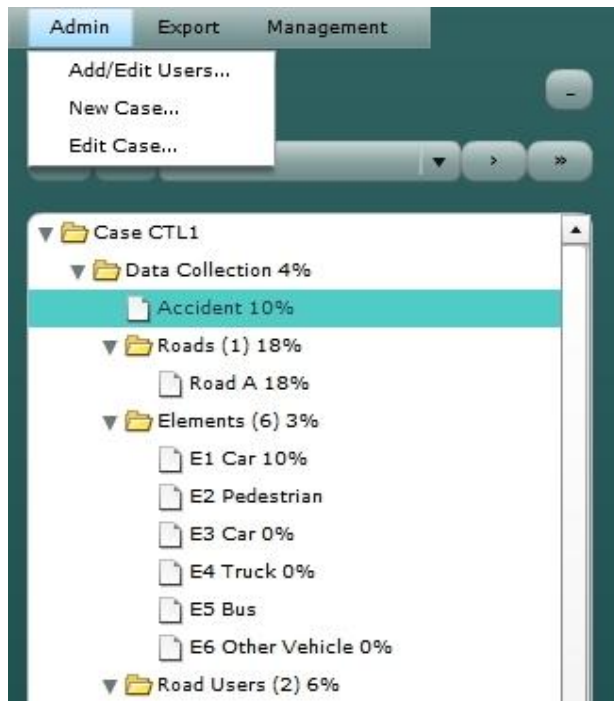


Figure 7-4 Admin menu

7.3.1. Scenario: Accessing the user administration

Given user is logged in as operational manager or administrator.

- User clicks Admin on the top menu.

System displays menu alternatives *Users* *Cases*.

- User clicks *Users*.

System displays an Users-window which contains the Name, Roles and Edit button for each user and an “Add user” button.

7.3.2. Scenario: Accessing user administration with insufficient permission

Given user is logged in as something other than operational manager or administrator. Then system disables the “User administration” menu.

7.3.3. Scenario: Adding a new Case

Given user is logged in with valid credentials.

Alternative A1 (if user is logged in as investigator/team organizer/operations manager/administrator):

- When user clicks Admin->New Case on the main menu.

The system then displays a “New Case” window,

User tabs thru:

- Center, selects value from a list.
- Start, selects a date.
- End, selects a date.
- Comment, enters a text.

Alternative B1 – Add Users to Case

- User selects Users from a list that consists of a list of users that are not blocked or expired and clicks the “Add” button.

The system then inserts the user in a grid.

User selects one or more case roles for the added user (Member, Leader, Medical Investigator).

Alternative B2 – Remove User from Case

When the user selects a role in the grid and clicks the “Remove” button.

- The system then removes the role from the grid.

When the user has entered values in all required fields.

- The system then enables the “Save” button.
- The user “clicks” the “Save” button.
- Then system creates a new Case with status “Work in progress” and Accident and adds the Case in the “Case Tree”. The new Case gets a new case ID based on “Case Center” and an number (e.g. GBG2261).

Alternative A2 (If user is logged in as internal analyst/external analyst).

System has disabled the Admin->Cases->New menu.

In Figure 7-5 a view of the form for adding a new case is reported.

New Case

Details

Center: CTL

Start: [Date Picker]

End: [Date Picker]

Comment: [Text Area]

Case Members

Roberto Carroccia [Add]

User	Member	Leader	Medical Investigator

Save

Figure 7-5 New Case Form

7.3.4. Scenario: Edit a Case

Given user is logged in with valid credentials.

- Alternative A1 (if user is logged in as Administrator/Operational Manager/Team Organizer or is Case Leader on the selected case):
- When user clicks Admin->Edit Case on the main menu.

The system then displays “Edit Case” window.

- User can edit the case values (See scenario “Adding a new Case”).
- User can change the Status of a Case, selects value from a list (See scenario “Change Case” status).
- When user has entered values in all required fields.

The system then enables the “Save” button.

- When user “clicks” the “Save” button.

System saves the current Case.

- Alternative A2 (If user is logged in as Investigator/Internal Analyst/External Analyst):

System has disabled the Admin->Cases->Edit menu.

7.3.5. Scenario: Change Case Status

Given user is logged in with valid credentials.

- When user clicks Admin->Cases->Edit on the main menu.

The system then displays “Edit Case” window.

- Alternative A - Change to “Published”
- User selects “Published” from the Status combo-box and clicks “Save”.

System changes the Status to “Published”

- Alternative B – Change to “Finished”
- User selects “Finished” from the Status combo-box and clicks “Save”.

System changes the Status to “Finished”

- Alternative C – Change to “Work in progress”
- User selects “Work in progress” from the Status combo-box and clicks “Save”.

System changes the Status to “Work in progress”

In Figure 7-6 a view of the form for editing a case is reported.

User	Member	Leader	Medical Investigator	
Massimo Robibaro	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Remove
Roberto Carroccia	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Remove

Figure 7-6 Edit Case Form

7.4. Input Accident data

The screenshot shows the 'Accident Spot' tab of the 'Case CTL1 - Accident' system. The interface is divided into several sections for data entry:

- Administration:** Includes a dropdown for 'Type of road investigation' set to '2 | On scene'.
- Administrative On-Scene Variables:** Contains fields for 'Time of alarm' (12:32), 'Time of arrival' (13:02), 'Team back at office' (20:40), and 'Accessibility to the accident scene' (2 | Good).
- Administrative Retrospective Variables:** Contains fields for 'Road inspection duration [min]' (Not applicable), 'Road inspection date' (7777), and 'Time to drive to (one way) accident spot [min]' (Not applicable).
- On-Scene Investigators:** A list of members including Massimo Robibaro, Roberto Carroccia, and Gabriele Giustinia.
- Road Investigators:** A list of members.

Figure 7-7 Accident data section

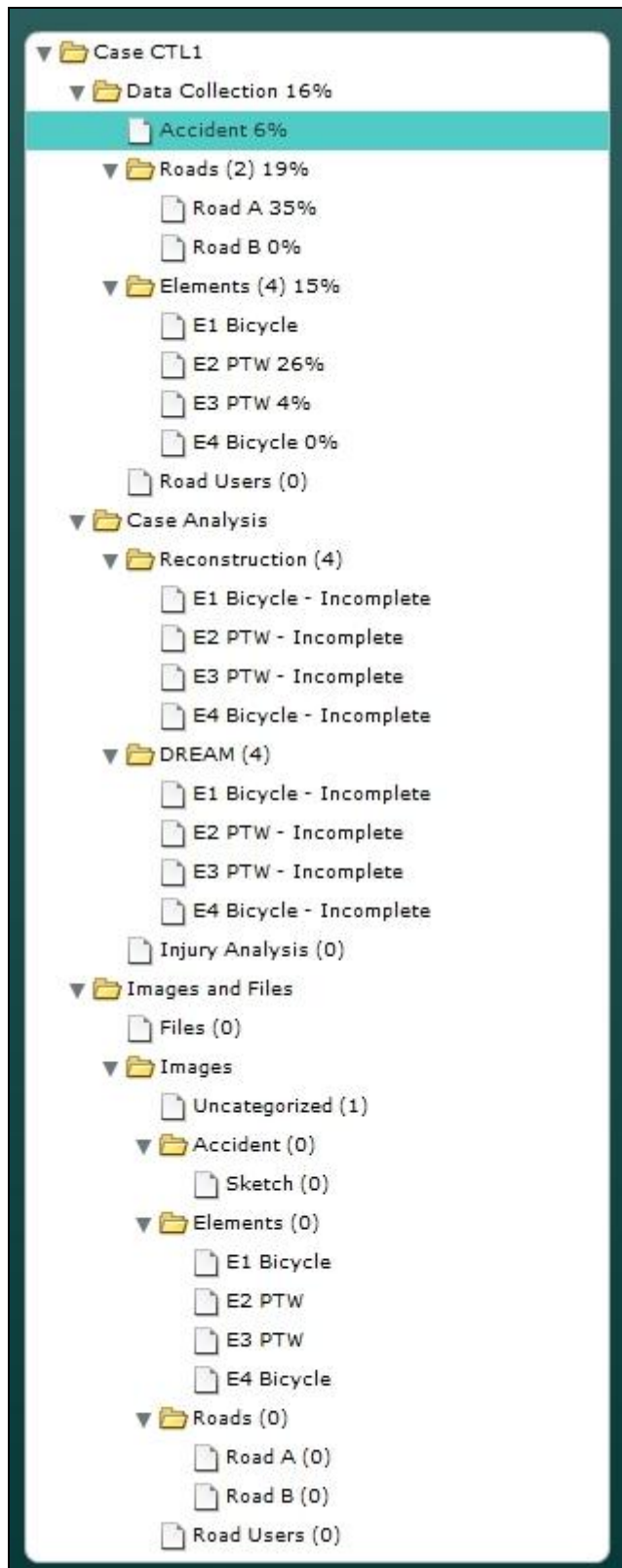


Figure 7-8 Accident Case Tree

7.4.1. Scenario: Input Accident Data

Given user is logged in as Internal Investigator/Team Organizer/Operations Manager

User selects a Case that is not published in the Case tree, navigates to the Accident node and clicks the Accident Node.

System displays the Accident View with an

- Admin tab
- General tab
- Summary tab
- Position tab
- Weather tab
- Witness tab
- Accident Spot tab



Figure 7-9 Tabs names under Accident section – first level

In the tabs, the variables are displayed and the related value can be inserted. On the Wiki, clear descriptions of the variables are reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.5. Input Car Data

Case CTL1 - E5 Car

Administration General Impacts Exterior Interior EDR Safety Systems

Inspection completed [dropdown] ?

Inspection date [calendar icon] ?

Inspection duration [min] [text input] ?

Source of information locating vehicle [dropdown] ?

Distance to inspection site [km] [text input] ?

mandatory M.O.T. inspection done? [dropdown] ?

Date of the M.O.T.(month, year) [text input] ?

Investigators

Member [dropdown]

[5 empty rows]

Figure 7-10 Car data section

7.5.1. Scenario: Create a Car object

Given user is logged in, has selected a Case in the Case Navigation Panel

- When user “right clicks” on the Element node in the Case tree.

The system then displays a pop up menu with the text “Add car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

When user clicks “Add Car”

The system then creates a Car object and inserts a new Car leaf under the Element node. The System checks out the created Car and opens the Car view with:

- Administration tab
- General tab
 - 1 tab
 - 2 tab
- Impacts tab
 - General tab
 - Frontal tab
 - Rear tab

- Left side tab
- Right side tab
- Top tab
- Pedestrian tab
- Exterior tab
 - General tab
 - Doors and Glazing tab
 - Wheels tab
 - Trailers tab
- Interior tab
 - General tab
 - Belt and Seat tab
 - Airbag tab
 - Interior observations tab
- EDR tab
- Safety systems tab



Figure 7-11 Tabs names under Car section – first level

- The system displays the Admin view of the selected Car and displays the element id and type (eg. E1 Car) of the selected Car in the Info Panel above the tab view.

In the tabs, the variables are displayed and the related value can be inserted. On the Online Manual, clear descriptions of the variables are reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.5.2. Scenario: Delete a Car object

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Car leaf under the Element node in the Case tree.

- The system then displays a pop up menu with the text “Delete this Car”.

When user clicks “Delete this Car”

- The system then displays a pop up message: “Are you sure you want to delete Car XX”, with a “OK” and “Cancel” button.

Alt A1

- When user selects “Cancel”

The system does not delete the Car and does not remove the Car leaf from the Case tree.

- Alt A2
- When user selects “Ok”

The system deletes the Car and removes the Car leaf from the Case tree.

7.5.3. Scenario: Input Car data – Admin Tab

Given user is logged in, has checked out the selected Car and selected the Admin Tab on the Car form.

User tabs thru:

- Inspection completed, selects a value from a list
- Inspection date, selects date from a date selector or enters a date.
- Inspection duration, enters a number
- Source of information locating the vehicle, selects a value from a list
- Distance to inspection site, enters a number.
- Alt A1 - Add Investigators
 - User selects Investigators from a list that consists of the Case members and clicks the “Add” button.
 - (An investigator can only be added once)
 - **The system inserts the name of the investigator in a grid.**
- Alt A2 – Remove Investigator.
 - When user selects an investigator in the grid and clicks the “Remove” button.
 - **The system removes the investigator from the grid.**

7.6. Input Bus Data

Case CTL1 - E6 Bus

Administration General Impacts Exterior Interior Safety Systems

Inspection completed [dropdown] ?

Inspection date [calendar icon] ?

Inspection duration [min] [input] ?

Source of information locating vehicle [dropdown] ?

Distance to inspection site [km] [input] ?

Investigators

Member

Figure 7-12 Bus data section

7.6.1. Scenario: Create a Bus object

Given user is logged in, has selected a Case in the Case Navigation Panel

- When user “right clicks” on the Element node in the Case tree.

The system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

- When user clicks “Add Bus”

The system creates a Bus object and inserts a new Bus leaf under the Element node. The System checks out the created Bus and opens the Bus view with:

- Administration tab
- General tab
- Impacts tab
 - General tab
 - Deformations tab
 - Pedestrian tab
- Exterior tab
 - General tab
 - Doors and Glazing tab
 - Wheels tab

- Interior tab
 - General tab
 - Belt and Seat tab
 - Airbag tab
 - Interior observations tab
- Safety systems tab



Figure 7-13 Tabs names under Bus section – first level

- The system displays the Admin view of the selected Bus and displays the name of the selected Bus in the Info Panel above the tab view.

Within the tabs, the variables are displayed and the related value can be inserted. On the Online Manual, a clear description of the variables are reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.6.2. Scenario: Delete Bus

Given user is logged in, has selected a Case in the Case Navigation Panel

- When user selects and “right clicks” on a Bus leaf under the Element node in the Case tree.

The system displays a pop up menu with the text “Delete Bus”.

- When user clicks “Delete Bus”

The system displays a pop up message with “OK” and “Cancel” buttons.

- Alt A1
- When user selects “Cancel”

The system does not delete the Bus and does not remove the Bus leaf from the Case tree.

- Alt A2
- When user selects “Ok”
- The system deletes the Bus and removes the Bus leaf from the Case tree.

7.6.3. Scenario: Input Bus Data – Admin Tab

Given user is logged in, has checked out the selected Bus and selected the Admin Tab on the Bus form.

User tabs thru:

- Inspection completed, selects a value from a list
- Inspection date, selects date from a date selector
- Inspection duration, enters a number

- Source of information locating the vehicle, selects a value from a list
- Distance to inspection site, enters a number.
- Alt A1 - Add Investigators
 - User selects Investigators from a list that consists of the Case members and clicks the “Add” button.
 - (An investigator can only be added once)

The system inserts the name of the investigator in a grid.

- Alt A2 – Remove Investigator.
 - When user selects an investigator in the grid and clicks the “Remove” button.

The system displays a pop up message: “Deleting Intact User”, with a “OK” and “Cancel” button

- Alt A2.1
 - When user selects “Cancel”
 - The system does not delete the Investigator and does not remove the Investigator from the grid.
- Alt A2.2
 - When user selects “Ok”
 - The system removes the investigator from the grid.

7.7. Input Truck Data

Case CTL1 - E7 Truck

Administration General Impacts Exterior Interior Safety Systems

Inspection completed [dropdown] ?

Inspection date [calendar icon] ?

Inspection duration [min] [text input] ?

Source of information locating vehicle [dropdown] ?

Distance to inspection site [km] [text input] ?

Investigators

Member

Figure 7-14 Truck data section

7.7.1. Scenario: Create a Truck object

Given user is logged in, has selected a Case in the Case Navigation Panel

- When user “right clicks” on the Element node in the Case tree.

The system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

- When user clicks “Add Truck”

The system creates a Truck object and inserts a new Truck leaf under the Element node. The System checks out the created Truck and opens the Truck view with:

- Administration tab
- General tab
- Impacts tab
 - General tab
 - Pillar tab
 - Roof tab
 - Cab tab
 - Underrun protection tab
 - Pedestrian tab
- Exterior tab
 - General tab
 - Geometry tab
 - Doors and glazing tab
 - Trailer tab
 - Wheels tab
 - Cargo and weight tab
- Interior tab
 - General tab
 - Belt & Seat tab
 - Airbag tab
 - Interior observations tab
- Safety systems tab



Figure 7-15 Tabs names under Truck section – first level

- The system displays the Admin view of the selected Truck and displays the name of the selected Truck in the Info Panel above the tab view.

In the tabs, the variables are displayed and the related value can be inserted. On the Online Manual clear descriptions of the variables are reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.7.2. Scenario: Delete Truck

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Truck leaf under the Element node in the Case tree.

The system displays a pop up menu with the text “Delete Truck”.

When user clicks “Delete Truck”

The system displays a pop up message with “OK” and “Cancel” buttons.

- Alt A1
- When user selects “Cancel”

The system does not delete the Truck and does not remove the Truck leaf from the Case tree.

- Alt A2
- When user selects “Ok”
- The system deletes the Truck and removes the Truck leaf from the Case tree.

7.7.3. Scenario: Input Truck Data- Admin Tab

Given user is logged in, has checked out the selected Truck and selected the Admin Tab on the Truck form.

User tabs thru:

- Inspection completed, selects a value from a list
- Inspection date, selects date from a date selector
- Inspection duration, enters a number
- Source of information locating the vehicle, selects a value from a list
- Distance to inspection site, enters a number.
- Alt A1 - Add Investigators
 - User selects Investigators from a list that consists of the Case members and clicks the “Add” button.
 - (An investigator can only be added once)
 - **The system inserts the name of the investigator in a grid.**
- Alt A2 – Remove Investigator.
 - When user selects an investigator in the grid and clicks the “Remove” button.
 - **The system displays a pop up message: “Deleting Intact User”, with a “OK” and “Cancel” button**
 - Alt A2.1

- When user selects “Cancel”
- **The system does not delete the Investigator and does not remove the Investigator from the grid.**
- Alt A2.2
- When user selects “OK”
- **The system removes the investigator from the grid.**

7.8. Input Other Vehicle

Case CTL1 - E8 Other Vehicle

General Impacts

Vehicle identification

Registration Number ?

Traveled Lane ?

General

Vehicle type ?

Kerb weight [Kg] or estimated weight ?

Dominant colour ?

Conspicuity ?

Motorcycle equipped with side car ?

Sitting height [mm] ?

Figure 7-16 Other Vehicle data section

7.8.1. Scenario: Create an Other Vehicle object

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Element node in the Case tree.

The system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

When user clicks “Add Other Vehicle”

The system creates an Other Vehicle object and inserts a new Other Vehicle leaf under the Element node. The System checks out the created Other Vehicle and opens the Other Vehicle view:

- General tab
- Impacts tab



Figure 7-17 Tabs names under Other Vehicle section – first level

- The system displays the selected Other Vehicle and displays the name of the selected Other Vehicle in the Info Panel above the tab view.

Within the tabs, the variables are displayed and the related value can be inserted. On the Online Manual, clear descriptions of the variables are reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.8.2. Scenario: Delete Other Vehicle

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on an Other Vehicle leaf under the Element node in the Case tree.

The system displays a pop up menu with the text “Delete Other Vehicle”.

When user clicks “Delete Other Vehicle”

The system displays a pop up message with “OK” and “Cancel” buttons.

Alt A1

- When user selects “Cancel”
- **The system does not delete the Other Vehicle and does not remove the Other Vehicle leaf from the Case tree.**

Alt A2

- When user selects “Ok”
- The system deletes the Other Vehicle and removes the Other Vehicle leaf from the Case tree.

7.8.3. Scenario: Input Other Vehicle Data – General Tab

Given user is logged in, has checked out the selected the general tab on Other Vehicle from.

User tabs thru Vehicle identification panel:

- Registration number, enters a text. Only visible for investigators.
- In which lane was the element driven, selects value from a list.

User tabs thru General panel:

- Vehicle type, selects value from a list.
- Kerb weight [Kg] or estimated weight, enters a number.
- Dominant colour, selects value from a list.
- Conspicuity, selects value from a list.

- Motorcycle equipped with side car, selects value from a list. Enabled if “Vehicle type “ is “Motorcycle” or “Other”
- Bicycle sitting height [mm], enters a number. Enabled if “Vehicle type “ is “Bicycle” or “Other”.

7.9. Input Pedestrian Data



Figure 7-18 Pedestrian data section

7.9.1. Scenario: Create a Pedestrian

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Element node in the Case tree.

Then system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

When user clicks “Add Pedestrian”

The system creates a Pedestrian object and inserts a new Pedestrian leaf under the Element node. The System checks out the created Pedestrian and opens the Pedestrian view:

- Impacts tab

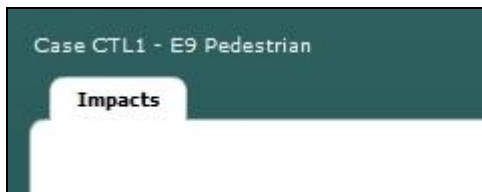


Figure 7-19 Tabs names under Pedestrian section – first level

- The system displays the selected Pedestrian and displays the name of the selected Pedestrian in the Info Panel above the tab view.

In the tabs the variables are displayed and the related value can be inserted. On the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.9.2. Scenario: Delete Pedestrian

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Pedestrian leaf under the Element node in the Case tree.

The system displays a pop up menu with the text “Delete Pedestrian”.

When user clicks “Delete Pedestrian”

The system displays a pop up message with “OK” and “Cancel” buttons.

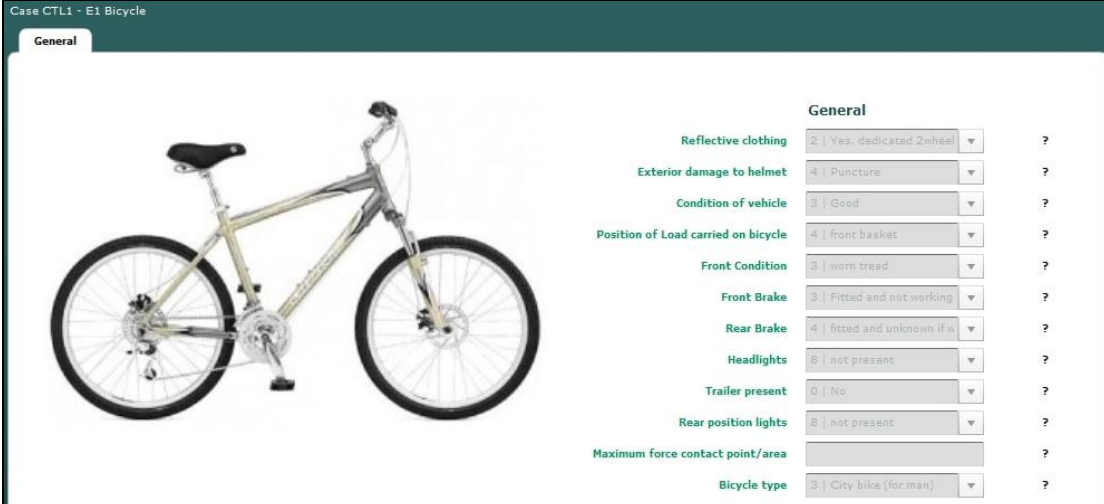
Alt A1

- When user selects “Cancel”
- **The system does not delete the Pedestrian and does not remove the Pedestrian leaf from the Case tree.**

Alt A2

- When user selects “Ok”
- The system deletes the Pedestrian and removes the Pedestrian leaf from the Case tree.

7.10. Input Bicycle Data



Case CTL1 - E1 Bicycle

General

Reflective clothing	2 Yes, dedicated 2wheel	?
Exterior damage to helmet	4 Puncture	?
Condition of vehicle	3 Good	?
Position of Load carried on bicycle	4 front basket	?
Front Condition	3 worn tread	?
Front Brake	3 Fitted and not working	?
Rear Brake	4 fitted and unknown if w	?
Headlights	8 not present	?
Trailer present	0 No	?
Rear position lights	8 not present	?
Maximum force contact point/area		?
Bicycle type	3 City bike (for man)	?

Figure 7-20 Bicycle data section

7.10.1. Scenario: Create a Bicycle

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Element node in the Case tree.

The system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

When user clicks “Add Bicycle”

The system creates a Bicycle object and inserts a new Bicycle leaf under the Element node. The System checks out the created Bicycle and opens the Bicycle view:

- General tab



Figure 7-21 Tabs names under Bicycle section – first level

The system displays the selected Bicycle and displays the name of the selected Bicycle in the Info Panel above the tab view.

In the tabs the variables are displayed and the related value can be inserted. On the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.10.2. Scenario: Delete Bicycle

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Bicycle leaf under the Element node in the Case tree.

The system displays a pop up menu with the text “Delete Bicycle”.

When user clicks “Delete Bicycle”

The system displays a pop up message with “OK” and “Cancel” buttons.

Alt A1

- When user selects “Cancel”
- **The system does not delete the Bicycle and does not remove the Bicycle leaf from the Case tree.**

Alt A2

- When user selects “Ok”
- The system deletes the Bicycle and removes the Bicycle leaf from the Case tree.

7.11. Input PTW Data

The screenshot displays the 'PTW data section' for 'Case CTL1 - E2 PTW'. It features three tabs: 'General', 'Wheel', and 'Safety Systems'. The 'Wheel' tab is selected. The form is organized into three main sections:

- Make and model:** Includes fields for Make (8318), Model, Variant, Colour, Year of manufacture (2008), PTW - Condition of vehicle (3 | Good), Motorcycle type (3 | road race replica), Motorcycle side car (2 | None), Side Car Damaged in Crash (0 | No), PTW - Position of Load carried on Motorcycle (3 | handle bar bags), Motorcycle Odometer reading, and Motor displacement.
- Mechanical parts:** Includes Anti-dive suspension system fitted (3 | fitted) and Drive train status (3 | worn).
- Other:** Includes Windscreen (4 | not present), Windscreen Damaged in Crash (0 | No), PTW - Headlights (4 | present, original equip), Automatic Headlights On (AHO) equipped (8 | not present), Daytime Running Lights (DRL) Equipped, Adaptive headlight fitted (0 | No), Front indicators, PTW - Rear position light (4 | present, original equip), Rear indicators, O/S mirror, Left mirror operational (1 | Yes), Left mirror damaged in crash, N/S mirror (3 | present, after market), Right mirror operational (1 | Yes), and Right mirror damaged in crash.

Figure 7-22 PTW data section

7.11.1. Scenario: Create a PTW

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Element node in the Case tree.

Then system displays a pop up menu with the text “Add Car”, “Add Truck”, “Add Bus”, “Add Other Vehicle” or “Add Pedestrian” or “Add bicycle” or “Add PTW”

When user clicks “Add PTW”

Then system creates a PTW object and inserts a new PTW leaf under the Element node. The System checks out the created PTW and opens the PTW view:

- General tab
- Wheel tab
- Safety systems tab
 - PPE (Personal protective equipment) tab
 - Brake and handling system tab



Figure 7-23 Tabs names under PTW section – first level

The system displays the selected PTW and displays the name of the selected PTW in the Info Panel above the tab view.

In the tabs the variables are displayed and the related value can be inserted. On the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.11.2. Scenario: Delete PTW

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a PTW leaf under the Element node in the Case tree.

Then system displays a pop up menu with the text “Delete PTW”.

When user clicks “Delete PTW”

Then system displays a pop up message with “OK” and “Cancel” buttons.

Alt A1

- When user selects “Cancel”
- **Then system does not delete the PTW and does not remove the PTW leaf from the Case tree.**

Alt A2

- When user selects “Ok”
- Then system deletes the PTW and removes the PTW leaf from the Case tree.

7.12. Input Road Data

Figure 7-24 Road data section

7.12.1. Scenario: Create Road Object

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Road node in the Case tree.

Then system displays a pop up menu with the text “Add a new road”.

When user clicks “Add a new road”

Then system creates a Road object and inserted under the Road node in the Case tree and displays the Road view with:

- General tab
- Vulnerable Road User tab
- Road area tab
 - Road design tab
 - Barrier tab
 - Lane tab
 - Road side tab
- Collision Objects tab



Figure 7-25 Tabs names under Road section – first level

In the tabs the variables are displayed and the related value can be inserted. On the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.12.2. Scenario: Delete Road Object

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Road in the Case tree.

Then system displays a pop up menu with the text “Delete this Road”.

When user clicks “Delete this Road”

Then system displays a pop up message: “Are you sure you want to delete this Road?” with an “OK” and “Cancel” button.

Alt A1

- When user selects “Cancel”
- **Then system does not delete the Road and does not remove it from the Case tree.**

Alt A2

- When user selects “OK”
- **Then system deletes the Road and removes the Road from the Case tree**

7.12.3. Scenario: Input Road Data

Given user is logged in and has selected a Case in the Case tree Navigation Panel

When user selects a Road in the Case tree

Then system displays the name of the selected Road in the information bar and displays the General tab in the Road form

7.13. Input Road User Data

The screenshot shows a web application interface for 'Case CTL1 - Road User 1'. At the top, there are several tabs: 'Individual data', 'Trip Specific', 'PRS', 'Rescue', 'Medical Outcome', 'Injuries', and 'LTI - 1 m'. Below these, there are three sub-tabs: 'General Information', 'Anthropometric Data', and 'Health'. The 'General Information' sub-tab is selected and displays a form with the following fields:

Field Name	Input Type	Required?
Identification...	Text input	?
Age	Text input	?
Age in months	Text input	?
Estimated Age	Dropdown menu	?
Gender	Dropdown menu	?
Pregnancy	Dropdown menu	?
Name	Text input	?
Phone	Text input	?
Address	Text input	?

Figure 7-26 Road User data section

7.13.1. Scenario: Create a Road User

Given user is logged in, has selected a Case in the Case Navigation Panel

When user “right clicks” on the Road User node in the Case tree.

Then system displays a pop up menu with the text “Add a new Road User”.

When user clicks “Add a new Road User”

Then system creates a Road User object and inserts a new Road User leaf under the Road User node. The System checks out the created Road User and opens the Road User view with:

- Individual data tab
 - General information tab
 - Anthropometric data tab
 - Health tab
- Trip Specific tab
- PRS(Passive Restraining Systems) tab
- Rescue tab

- General tab
- Treatment tab
- Medical outcome tab
- Injuries tab
- LTI – 1 m (Long term injuries follow up – 1 month) tab
 - Administration tab
 - Complaints tab
 - General complaints tab
 - Physical complaints tab
 - Work absence tab
 - Care tab
- LTI – 6 m (Long term injuries follow up – 6 month) tab
 - Administration tab
 - Complaints tab
 - General complaints tab
 - Physical complaints tab
 - Work absence tab
 - Care tab
- LTI – 12 m (Long term injuries follow up – 12 month) tab
 - Administration tab
 - Complaints tab
 - General complaints tab
 - Physical complaints tab
 - Work absence tab
 - Care tab
- LTI – 24 m (Long term injuries follow up – 24 month) tab
 - Administration tab
 - Complaints tab
 - General complaints tab
 - Physical complaints tab
 - Work absence tab
 - Care tab

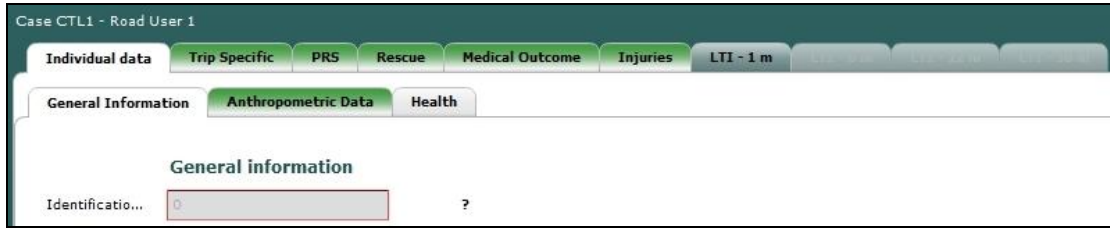


Figure 7-27 Tabs names under Road User section – first level

The System displays the selected Road User id, which Traveled Element and Seat Position (if applicable) the Road User belongs to.

In the tabs the variables are displayed and the related value can be inserted. On the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.13.2. Scenario: Delete a Road User

Given user is logged in, has selected a Case in the Case Navigation Panel

When user selects and “right clicks” on a Road User leaf under the Road User node in the Case tree.

Then system displays a pop up menu with the text “Delete this Road User”.

When user clicks “Delete this Road User”

Then system displays a pop up message: “Are you sure you want to delete this Road User”, with a “OK” and “Cancel” button.

Alt A1

- When user selects “Cancel”
- **Then system does not delete the Road User and does not remove the Road User leaf from the Case tree.**

Alt A2

- When user selects “Ok”
- **Then system deletes the Road User and removes the Road User leaf from the Case tree.**

7.14. Input DREAM

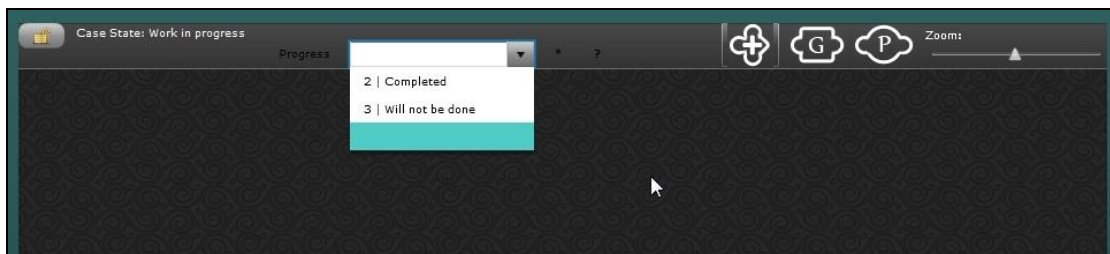


Figure 7-28 DREAM data section

7.14.1. Scenario: Create a DREAM Analysis

Given user is logged in, has selected a Case in the Case Navigation Panel

When user creates an “Element”.

Then system creates a DREAM object and inserts a new DREAM leaf under the DREAM Node in the Case tree.

For further information please have a look at the DREAM manual.

7.14.2. Scenario: Delete a DREAM Analyses

Given user is logged in, has selected a Case in the Case Navigation Panel

When user deletes an “Element”.

Then system deletes the corresponding DREAM Analysis and removes the DREAM Analyses leaf from the Case tree.

7.14.3. Scenario: Create a Chart

User checks out the selected DREAM Analysis.

Then system enables the “Chart comment”, “genotype” and “phenotype” symbols.

7.15. Input Reconstruction

The screenshot displays a web interface for 'Case CTL1 - E7 Truck'. It features a 'Vehicle Impact' tab and a 'Reconstruction Progress' dropdown menu. The main content area is divided into two columns: 'Event' and 'Rollover'. The 'Event' column contains several dropdown menus and input fields, including 'Event type', 'Occurred during rollover', 'Impact', 'Collision partner', 'Road collision object', 'Collision partner event', 'Event time in [s]', and 'Event distance in [m]'. The 'Rollover' column contains dropdown menus and input fields for 'Initial rollover direction', 'Number of lateral rolls', 'Number of longitudinal rolls', 'Rollover type', and 'Rollover causing event'. A large text area for 'Description of rollover initiation' is located at the bottom right of the form.

Figure 7-29 Reconstruction data section - Truck example

7.15.1. Scenario: Create Reconstruction Object

When user creates an element object then system creates a corresponding reconstruction element of the same element type and adds a reconstruction element leaf under the Reconstruction node.

7.15.2. Scenario: Delete Reconstruction Object

When user deletes an element node then system deletes the corresponding reconstruction object and removes the reconstruction leaf from the case tree.

- Sequence tab
- Vehicle Impact tab

- Pedestrian Impact tab



Figure 7-30 Tabs names under Reconstruction section – first level - Truck

For further information please have a look on the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.16. Input Injury Analysis Data

A screenshot of a software interface for 'Case CTL1 - Road User 1'. The main section is titled 'Injury Analysis'. At the top left, there is a 'Progress' dropdown menu. Below it, the text 'Road User 1, E9 Pedestrian' is displayed. There are several input fields and dropdown menus: 'Injury related to event number' (dropdown with 'Not Selected'), 'Was the occupant part of the deformation zone' (checkbox), 'Injury mechanism' (large text area), 'Injury causing part' (dropdown), and 'Injury related to observation number' (dropdown with 'Not Selected'). Each of these fields has a question mark icon to its right.

Figure 7-31 Injury Analysis data section

7.16.1. Scenario: Create an Injury Analysis

Given user is logged in, has selected a Case in the Case Navigation Panel

When user creates a “Road User”.

Then system creates an Injury Analysis object and inserts a new Injury Analyses leaf under the Injury Analyses Node in the Case tree.

For further information please have a look on the Wiki a clear descriptions of the variables is reported (<http://dacota-investigation-manual-eu.ita.chalmers.se/>).

7.16.2. Scenario: Delete an Injury Analyses

Given user is logged in, has selected a Case in the Case Navigation Panel

When user deletes a “Road User”.

Then system deletes the corresponding Injury Analyses and removes the Injury Analyses from the Case tree.

7.17. Export

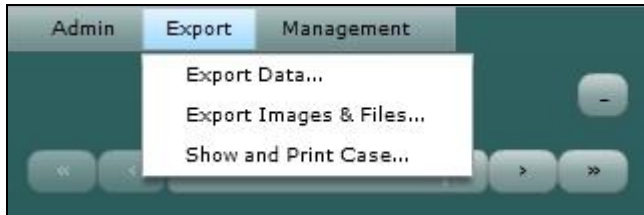


Figure 7-32 Export menu

7.17.1. Scenario: Export Data

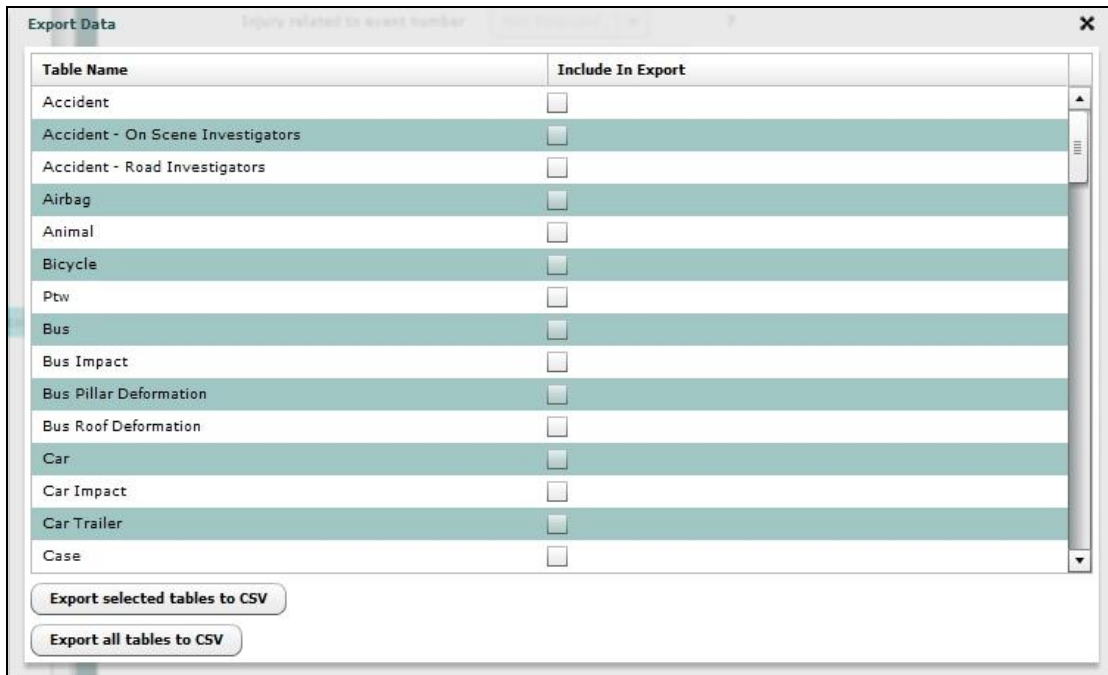


Figure 7-33 Export data section

When user selects “Export”->”Data” on the main menu

Then system displays the export data view with:

- Checkboxes for all tables in the database.
- “Export selected tables to CSV” button
- “Export all” button

Alt A1 – User exports data from all tables.

- User clicks the “Export all” button.

- **Then system creates one csv-files for each table in the database and crates a zip-file that's named "YYYYMMDD_full.zip" containing all csv-files.**

Alt A2 – User exports data from some tables

- User checks the checkboxes of the tables that should be exported and clicks the "Export to CSV"-button.
- **Then system creates a csv-file for each of the selected tables and creates a zip-file that's named "YYYYMMDD-partial.zip" containing the created csv-files.**

The csv-files contains the column names, data and a "." If no data exists .

The table "Intact value" is always exported in its entirety.

The following values are never exported:

- Road User Table
 - Identification code.
 - Name
 - Phone
 - Address
- Road Table
 - Police ref. number
- Car Table
 - Vehicle identification n
 - Registration n
- Bus Table
 - Vehicle identification n
 - Registration n
- Truck Table
 - Vehicle identification n
 - Registration n
- Other Table
 - Registration n

7.17.2. Scenario: Export Images and Files



Figure 7-34 Export images & files section

When user selects “Export”->”Images & Files” on the main menu

Then system displays the export data view with:

- “Download Images” button
- “Download Files” button
- “Export all” button

Alt A1 – User exports images from the current Case.

- User clicks the “Download images” button.
- **Then system creates a zip-file that’s named “CASE_###_IMAGES_YYYYMMDD.zip” containing all the images of the current case.**

Alt A2 – User exports files from the current Case.

- User clicks the “Download files” button.
- **Then system creates a zip-file that’s named “CASE_###_FILES_YYYYMMDD.zip” containing all the files of the current case.**

7.18. Handle images and upload files

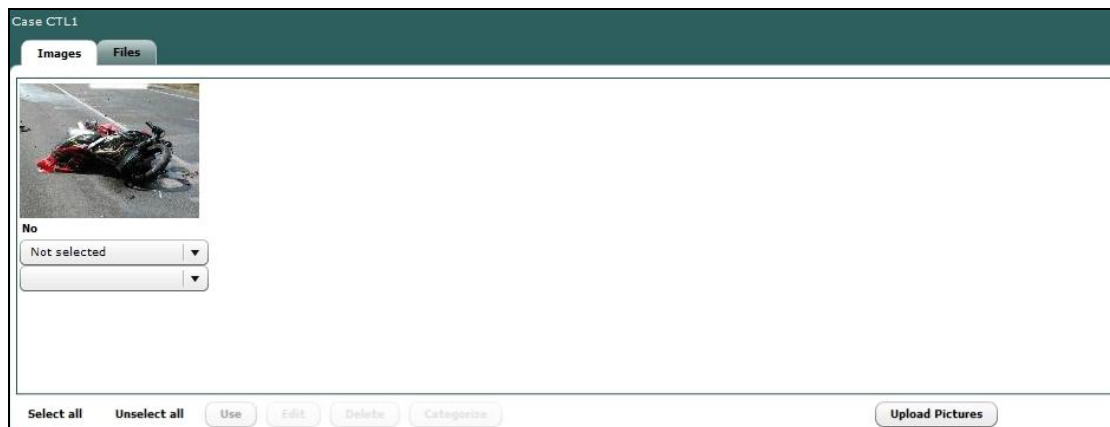


Figure 7-35 Images and Files management section

7.18.1. Scenario: Upload image

Given user is logged in, has selected a Case in the Case Navigation Panel.

When user clicks a node or leaf under the Image node in the Case tree.

Then system displays the Images view.

User clicks the “Browse” button.

Then system opens a browser window.

User selects the images to upload and clicks “open”.

Then system displays thumbnails of all the uploaded images and the images gets a name based on case id and a number, the images gets categorized as Uncategorized and can be located in the Case tree under the “Uncategorized” node.

7.18.2. Scenario: Edit image

Given user is logged in, has selected a Case in the Case Navigation Panel

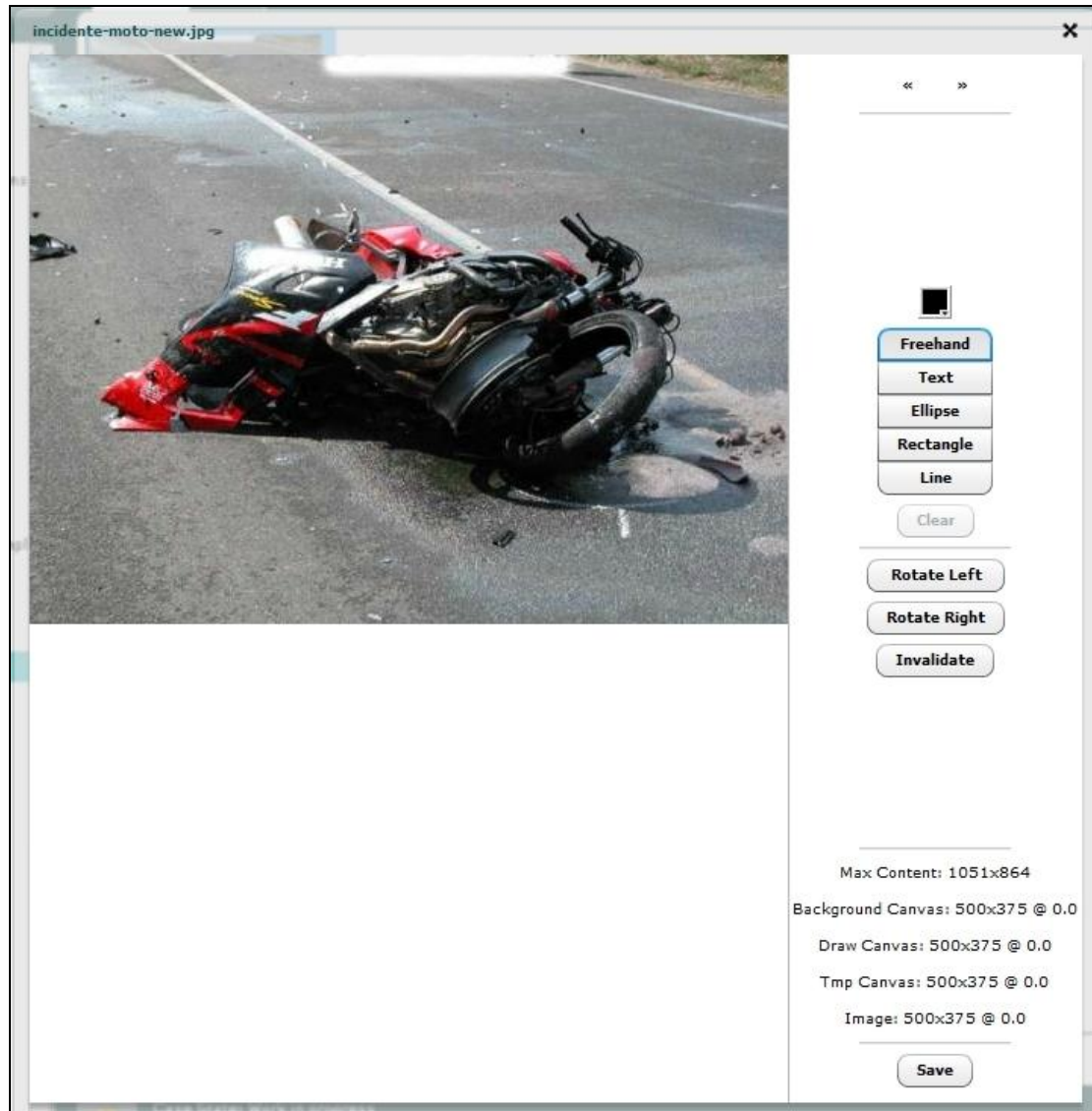


Figure 7-36 Image editing section

7.18.2.1. Scenario: Select images for editing

When user clicks a node or leaf under the Image node in the Case tree.

Then system opens the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

User checks the image/images to edit and clicks the “Edit” button.

Then system displays the edit view, with an image and editing tools.

7.18.2.2. Scenario: Edit freehand

User selects a color and clicks the “freehand” button.

Then system changes the cursor to a pen

User draws in the image.

7.18.2.3. Scenario: Edit Text

When user clicks “Edit Text”

Then the user can add some text to the image starting from where the mouse pointer has been clicked.

7.18.2.4. Scenario: Edit Ellipse

When user clicks “Edit Ellipse”

Then the user can add an ellipse to the image starting from where the mouse pointer has been clicked and ending where the mouse pointer is released.

7.18.2.5. Scenario: Edit Rectangle

When user clicks “Edit Rectangle”

Then the user can add a rectangle to the image starting from where the mouse pointer has been clicked and ending where the mouse pointer is released.

7.18.2.6. Scenario: Edit Line

When user clicks “Edit Line”

Then the user can add a line to the image starting from where the mouse pointer has been clicked and ending where the mouse pointer is released.

7.18.2.7. Scenario: Rotate left

When user clicks “Rotate left”

Then system rotates the image 90 degrees to the left.

7.18.2.8. Scenario: Rotate Right

When user clicks “Rotate right”

Then system rotates the image 90 degrees to the right.

7.18.2.9. Scenario: Save changes

When user clicks “Save changes”

Then system saves the new edit image and replaces the old image.

7.18.2.10. Scenario: Next image.

User clicks “Next image”. N/A if current image is the last of the selected images for editing.

Then system displays a new image that is the next of the selected images for editing.

7.18.2.11. Scenario: Previous image.

User clicks “Previous image”. N/A if current image is the first of the selected images for editing.

Then system displays a new image that is the previous of the selected images for editing.

7.18.2.12. Scenario: Return to category view.

User clicks “Back to picture category”.

Then system returns the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

7.18.3. Scenario: Delete Image

Given user is logged in, has selected a Case in the Case Navigation Panel.

When user clicks a node or leaf under the Image node in the Case tree.

Then system opens the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

User checks the image/images to delete and clicks the “Delete” button.

Then system deletes the selected images from the server.

7.18.4. Scenario: Categorize image

Given user is logged in, has selected a Case in the Case Navigation Panel.

When user clicks a node or leaf under the Image node in the Case tree.

Then system opens the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

User checks the image/images to categorize and clicks the “Categorize” button.

Then system displays a pop-up view.

User selects either a main category (Uncategorized, Accident, Element, Road or Road User) or a specific Element, Road or Road User.

7.18.5. Scenario: (Re)Name Image.

Given user is logged in, has selected a Case in the Case Navigation Panel.

When user clicks a node or leaf under the Image node in the Case tree.

Then system opens the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

User checks the image/images to rename and selects a value from the first combo box.

Alt A1 – User selects one of the following values:

- overview
- traces
- tyremark
- splinterfield
- leakage
- end position element no.
- reference no.

- element no.
- way to accident element no.
- collision object no.
- **Then user can select one of the following values in the second combo box:**
- the numbers 1 to 20
- all
- overview

Alt A2 - User selects one of the following values:

- front
- front/rh
- rh side
- rear/rh
- rear
- rear/lh
- lh side
- front/lh
- roof
- hood
- wheel
- door
- boot lid
- sunroof
- windscreen
- glazing
- deformation
- A-pillar
- B-pillar
- C-pillar
- D-pillar
- undercarriage
- other
- sill
- engine compartment
- battery
- bumper
- axle
- towing hook
- trailer

- trailer connection
- roof hatch
- Cab
- mirrors
- frame
- FUP
- SUP
- RUP
- fuel tank
- seatbelt
- airbag
- seat
- neck restraint
- dashpanel
- CRS
- trim
- headliner
- Luggage compartment
- Then user can select one of the following values in the second combo box:
 - 1.1
 - 1.2
 - 1.3
 - 2.1
 - 2.2
 - 2.3
 - 3.1
 - 3.2
 - 3.3
 - front
 - rear
 - LH
 - RH
 - front/RH
 - front/LH
 - rear/RH
 - rear/LH
 - axle-1

- axle-2
- axle-3
- axle-4

7.18.6. Scenario: Use in case report

Given user is logged in, has selected a Case in the Case Navigation Panel.

When user clicks a node or leaf under the Image node in the Case tree.

Then system opens the Images view and displays the thumbnails of the images that belong to the selected node/leaf.

7.18.6.1. Scenario: Add image for use in case report

User checks the image to use in case report and clicks the “Use in case report” button

Alt A1 – No other image is marked for use in case report.

- System marks the image with a “...” to indicate that it is used in case report.

Alt A2- Another image is marked for use in case report.

- Then system displays a pop-up “Another image is already used for case report”.

7.18.6.2. Scenario: Remove image from use in case report

User checks the image to remove from use in case report and clicks the “Use” in case report button

Alt A1 – The image is set as used in case report

- System displays a pop-up “Are you sure to remove the image ?”
- You have to answer Yes or Not

Alt A2 – The image is not set as used in case report

- See the scenario 7.18.6.1

7.19. Show and print Case

7.19.1. Create Case Report

When user selects “Export->Show and Print Case” on the main menu

Then system displays a pop-up with a “Create PDF-report button”

When user clicks the “Create PDF-report” button

Then system creates a pdf-file that’s named “case_XXXX_YYYY-MM-DD.pdf” (XXXX= the nr of the Case Id).

User selects a location to save the pdf-file.

7.19.2. Case Report Layout

7.19.2.1. Report Header

System adds the following data to a Case Report:

- Case ID
- Accident Date
- Accident Type
- Progress
- Accident summary
- Scene Sketch (if available)
- Scene Photo (if available)
- Elements involved in the accident (see below for detailed layout)
- Vehicle Photos (if available)
- Collision Object photos (if available)

7.19.2.2. Element of Vehicle type

System adds the following data to an Element of Vehicle type:

- A thumbnail picture of the element (the image that has been tagged as Use in report)
- Element Type
- Make
- Model
- Year and month of manufacture
- Color
- A table of all Events with:
 - Event id
 - Delta V
 - CDC code
 - Collision partner/Collision Object
- A table of all Road Users with the current element as “Traveled Element” with:
 - Seat position
 - Gender
 - Age
 - Belt Use
 - Max AIS
 - Dead

7.19.2.3. Element of Other Vehicle type

System adds the following data to an Element of Other Vehicle type:

- Element Type
- Vehicle Type
- A table of all Events with:
 - Event id
 - Delta V
 - Collision partner/Collision Object
- A table of all Road Users with the current element as “Traveled Element” with:

- Seat position
- Gender
- Age
- Type of helmet
- Max AIS
 - Dead

7.19.2.4. Element of Pedestrian type

System adds the following data to an Element of Pedestrian type:

- Element Type
- A table of all Events with:
 - Event id
 - Delta V
 - Collision partner/Collision Object
- A table of all Road Users with the current element as “Traveled Element” with:
 - Gender
 - Age
 - Max AIS
 - Dead

7.20. Comment

7.20.1. Scenario: Adding a comment on input fields

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

Then system displays the corresponding form with a star (“*”) after each input field. If the input field has a comment, the star is green.

When user clicks/presses “enter” on the star

Then system displays a pop-up window with a text area.

When user writes a comment.

Then system enables the “Save” button

Alt A1

- When user clicks “Save”
- **Then system saves comment, closes the pop-up window and colors the star green.**

Alt A2

- When user clicks “cancel”
- Alt A2.1 (no comment existed)
 - Then system closes the pop-up window and does not color the star.
- Alt A2.2
 - Then system leaves the old comment, closes the pop-up window and leaves the color on the star.

Scenario: Delete a comment on input fields

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

Then system displays the corresponding form with a star (“*”) after each input field. If the input field has a comment, the star is green.

When user clicks/presses “enter” on the star

Then system displays a pop-up window with a text area.

Alt A1 – A comment existed on the input field

- System has enabled the “Delete” button.
- When user clicks “Delete”
- Then system deletes the comment, closes the pop-up window and sets the color of the star back to black.

Alt A2 – No comment exists on the input field

- System has disabled the “Delete” button.
- When user clicks “cancel”
- Then system leaves the old comment, closes the pop-up window and leaves the color on the star.

7.21. Tool tip

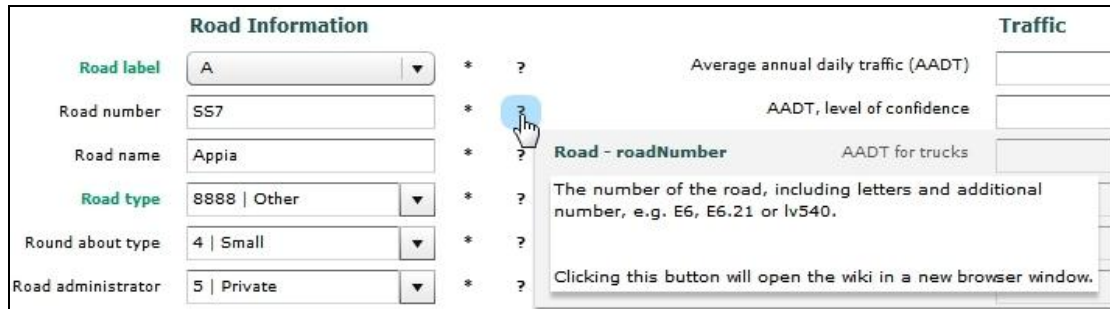


Figure 7-37 Tooltip with variable short description

7.21.1. Scenario: Read short description

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

Then system displays the corresponding form with a question mark (“?”) after each input field.

When user “hovers” with the mouse pointer over the question mark

Then system displays a pop up with the short description from the wiki.

7.21.2. Scenario: Go to the wiki

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

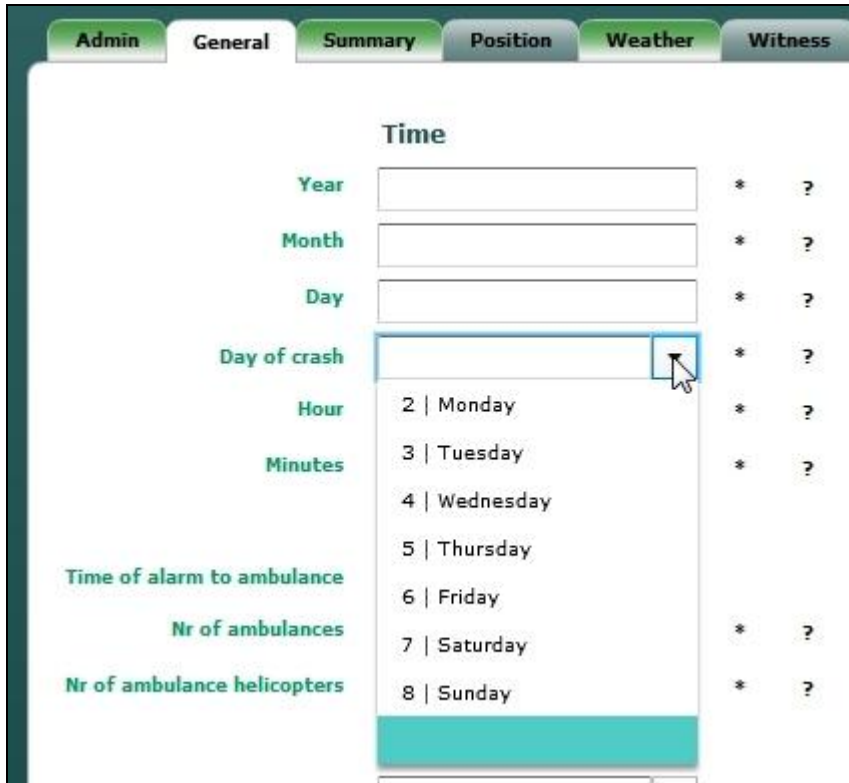
Then system displays the corresponding form with a question mark (“?”) after each input field.

When user clicks the question mark

Then system opens the wiki in a new browser window with the description for the selected variable.

7.22. Short keys

7.22.1. Scenario: Combo Box



The screenshot shows a web application interface with a navigation bar at the top containing tabs: Admin, General, Summary, Position, Weather, and Witness. The 'Summary' tab is active. Below the navigation bar, there is a form titled 'Time' with several input fields and a dropdown menu. The fields are: Year, Month, Day, Day of crash, Hour, Minutes, Time of alarm to ambulance, Nr of ambulances, and Nr of ambulance helicopters. The 'Day of crash' field is currently open, showing a list of days from Monday to Sunday. A mouse cursor is pointing at the dropdown arrow. To the right of each field, there are asterisks and question marks, indicating required fields and help links.

Field	Value	*	?
Year		*	?
Month		*	?
Day		*	?
Day of crash	2 Monday	*	?
Hour	3 Tuesday	*	?
Minutes	4 Wednesday	*	?
Time of alarm to ambulance	5 Thursday		
Nr of ambulances	6 Friday	*	?
Nr of ambulance helicopters	7 Saturday	*	?
	8 Sunday	*	?

Figure 7-38 Combo box example

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

Then system displays the corresponding form.

User sets focus on a Combo Box.

When user starts typing a number or a text.

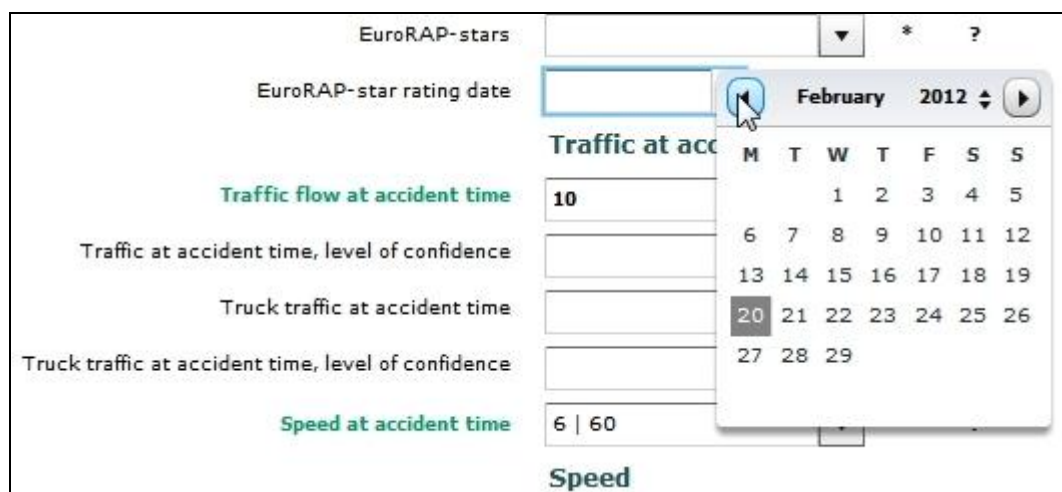
Then system filters the value list accordingly

If user presses ctr+arrow down the entire list is displayed.

The following values always means the same if they exists in the list:

- 0 always means “no”
- 1 always means “yes”
- 7777 always means “Not applicable”
- 8888 always means “Other”
- 9999 always means “Unknown”

7.22.2. Scenario: Text, Number or Date fields



The screenshot shows a web form with several input fields. A date calendar pop-up is displayed over the 'EuroRAP-star rating date' field. The calendar shows the month of February 2012, with the 20th selected. The form fields include:

- EuroRAP-stars (dropdown menu)
- EuroRAP-star rating date (text field with calendar pop-up)
- Traffic flow at accident time (text field with value 10)
- Traffic at accident time, level of confidence (text field)
- Truck traffic at accident time (text field)
- Truck traffic at accident time, level of confidence (text field)
- Speed at accident time (text field with value 6 | 60)
- Speed (text field)

Figure 7-39 Date calendar field component example

When user selects a case from the list of available cases in the “Case Navigation Panel” and selects a node.

Then system displays the corresponding form.

User sets focus on a Text or Number field.

The following values always means the same if the user types them in the Text or Number field:

- 7777 always means “Not applicable”
- 9999 always means “Unknown”.