

Software Development Concept and Report - 12/2013

Table of contents

1. INTRODUCING THE PROJECT "LEVEL5" TO EVALUATE INFORMAL LEARNING.....	2
PRELIMINARY REMARKS	2
1.1 MOTIVATION FOR THE EVALUATION OF INFORMAL LEARNING	2
1.2 Background:	2
1.3 Previous Software Support	4
1.3.1 GUI:	5
1.3.2 Design	5
1.3.3 Technical Requirements	6
2. CONSIDERATIONS OF A FURTHER DEVELOPMENT AND A RE-LAUNCH.....	7
2.1 New Requirements and Software Re-launch	7
2.2 Choice of Programming (Language)	7
2.3 General Considerations regarding the Design	7
2.4 Simplification of the model to produce a testable prototype	8
2.5 Prototype of a User Interface	9
2.6 Authentication, authorization, and other security goals	14
2.7 Single Sign On	14
2.8 Transaction security	15
3. OVERVIEW OF WORKING STEPS	16
4. GUIDED TOUR THROUGH THE SOFTWARE	20
4.1. Step 1: Manage learning project	20
4.2. Step 2: Inventory of competences	23
4.3. Step 3: Assessment Scenarios	26
4.4. Step 4: Validation	26
4.5. User Administration	29
5. ANNEXES.....	30
5.1. VITA-Certificate of a learner in a Turkish school.....	30
5.2. Learning Ticket from the learning festival in Berne	31

1. Introducing the project "LEVEL5" to evaluate informal learning

Preliminary Remarks

Through the planned introduction of ECVET Specification 2014, the need has arisen to combine the LEVEL5 software, or the underlying pedagogical approach with the ECVET specifications. To satisfy these and other new requirements, it was decided to re-develop the software from scratch. In order to achieve a loose coupling between the user interface and application logic the new software has been designed as a web service in the framework of the VITA project.

The project began in early 2012 and has a term of 2 years.

The development of the new software is described in the following three chapters:

1. Introduction to Project LEVEL5
2. Inventory / analysis of existing software and implications for adaptation
3. Redesign of a service-based software including a word-like description of the steps that are performed during the project.

When creating minimalist prototype software the frameworks JAAS and Rampart have been combined in order to achieve some major security goals. Rampart is an out-of-the-box solution that is essentially adapted over configuration files. JAAS, however, is primarily a design pattern for capability-based rights management, which forwards the Security Context requests about user rights to the interfaces defined by the framework.

1.1 Motivation for the Evaluation of Informal Learning

»Zunehmend legen Unternehmen Wert auf persönliche Fähigkeiten und Kompetenzen, die als Voraussetzung für die Ausübung jedweder beruflichen Tätigkeit als wichtig angesehen werden. Neben (fremd-)sprachlichen Fähigkeiten und Kompetenzen im Bereich ITC, stehen Soft-Skills wie Organisationsfähigkeit, Teamfähigkeit oder Anpassungsfähigkeit im Vordergrund. Das bedeutet auch, dass im Vergleich zu einem "Vorratslernen" in der Erstausbildung Lernformen immer wichtiger werden, die sich außerhalb der traditionellen Institutionen des Bildungssystems vollziehen. Erwerb und Anpassung von Kenntnissen, Fertigkeiten und Kompetenzen finden zunehmend auf non-formalen Wegen, in Weiterbildungsseminaren oder im selbst-organisierten Lernen statt. Gleichzeitig wächst die Bedeutung des informellen Kompetenzerwerbs, der sich ohne explizite Lernabsicht im praktischen Handeln, vor allem am Arbeitsplatz, aber auch bei vielen anderen Gelegenheiten in Freizeit, Familie oder Ehrenamt vollzieht.«¹

1.2 Background:

The focus of the project VITA is the development of an educationally and scientifically standardised method for describing service-oriented competences acquired through informal learning, and to connect it with a rating system that documents the progress in the acquisition of the tested skills. The assessments are carried out in 3 different dimensions.

¹ <http://www.ecvet.de/c.php/ecvetde/lernen/hintergrund/hintergrund.rsys>

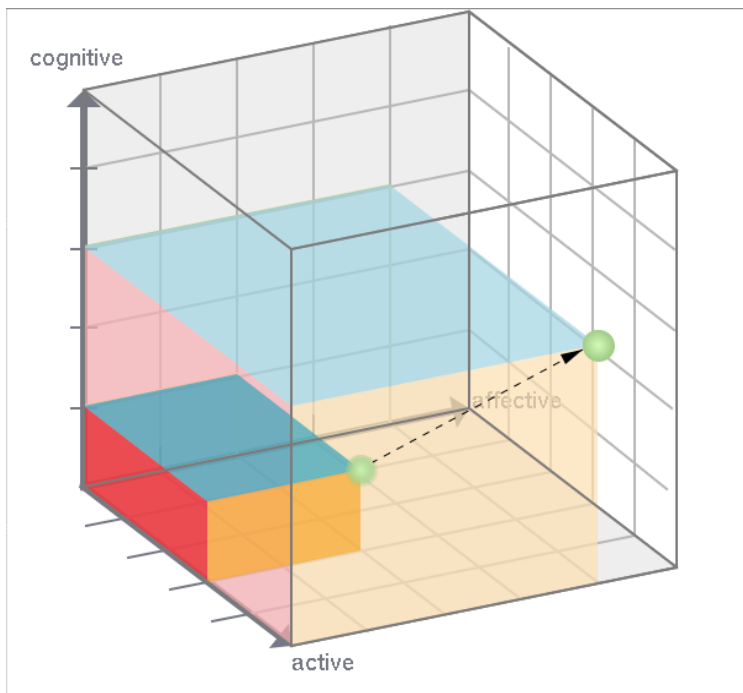
"Companies increasingly place value on personal skills that are seen as an essential prerequisite for the exercise of any professional activity. In addition to (foreign) language skills and expertise in ITC, there are soft skills such as organizational skills, teamwork and adaptability in the foreground. This also means that in comparison to a "supply learning" which takes place in the initial learning systems, learning outside the traditional institutions of the education system becomes increasingly important. Acquisition and adaptation of knowledge, skills and competencies are increasingly being held on non-formal ways in training seminars or self-organized learning. At the same time, the importance of informal skills acquisition increases, which takes place without explicit learning intention in practical action, especially in the workplace, but also on many other occasions in leisure, family or volunteering."

- Active Dimension: To what extent do the acquired competences show an impact on the activity level of the person?
- Cognitive dimension: To what extent are the acquired competences recognised in the classical sense cognitively?
- Affective dimension: To what extent have the acquired competences resulted in a change of the personal setting?

Each of these dimensions can be awarded by 1-5 points that describe its progress. The results of the assessment for the 3 dimensions are represented in a cube that shows a scale of 1-5 on each edge.

In order to make the results of the assessments of different groups of learners comparable, given templates (reference systems) are summarised in the so-called inventory.

If e.g. the development of social skills of a group of young people, who take part in an exchange programme, is assessed, a default template is created, which can be complemented by the evaluator, overwritten or fine-tuned in order to meet the individual requirements.



1.3 Previous Software Support

Since the main evaluation process is aiming to capture, modify and analyse data, it makes sense to replicate the procedure by software.

Since 2005 two web applications have been programmed for data collection and analysis:

- A first version of PHP. Basis for this was the cakephp framework (V1).
- A second Java JEE version (V2). Using JSF, Facelets (view), Hibernate (persistence), Batik (SVG Framework) and FOP (PDF). This version was developed by a team of partially up to five people in a period of two years. This version had been improved in the framework of VITA to give the partners the opportunity to learn how the software works during the first transnational training.

The version 3 of the LEVEL5 software will be finalised after approximately 18 development months (06/2013) and the practice test phase.

Further development of existing software

In the following, the need to extend / re-programming the software LEVEL5 (V 2) will be presented based on different requirements that stem from different user needs and use cases:

1.3.1 GUI:

Problems with 1 to n collection and presentation of data relationships.

- They occur in data collection time and again to 1 on n relations. An example: A person has a number of dates on which evaluations were made. The evaluations themselves are composed of different competence sets, which in turn are assigned in 3 different dimensions of 5 possible points. The GUI implementation is not systematic and consistent, which is confusing to the user.

Dealing with large amounts of data

- Large quantities of records are not yet presented clearly (e.g. in table form). Functions for sorting, searching, filtering, scrolling should be offered to support large data traffic.

1.3.2 Design

Monolithic design

- The complete application is a web application that relies on just one data source. The evaluations can only be made if people were created in the software. It is not possible to evaluate persons that are present in separate data sources (such as a personal database).

Strong coupling of the modules

- The application is programmed as the MVC pattern. However, there is a relatively strong coupling of the GUI consists to the business logic. The so called handler of the business layer can only handle events of logic, which are defined by the JSF framework. A connection of the business logic to another GUI (e.g. desktop or mobile GUI) was not intended and would be difficult to implement later.
- Hindered testability: The strong coupling between the GUI and business logic, the application is partially only testable via the GUI. Hence automated tests are difficult that are required for a comprehensive utilisation of the software in all envisaged educational sectors.

Rights management

- There are no rights or roles or groups of users. An evaluator accesses those records that she/he has created. Teamwork or read-only access to selected data sets is not possible.

Recycling and export of data

- The export of data or their recyclability on the platform in a different context is not provided.

Adaptability to other specifications

- The ECVET specification for the evaluation of educational qualifications in the EU is only slightly different from this software. However, the less well pronounced generic design and the strong coupling of the modules would make an adjustment to this specification rather difficult.

1.3.3 Technical Requirements

Hibernate Open-Session-In-View pattern

- For performance reasons the Hibernate Framework from V 2.0 used proxies for the entities. Only in the representation of the GUI these proxies will be filled with the actual data. In terms of a strict separation of MV, this is problematic, since the connection to the model during the rendering of the GUI must be maintained. Maintaining this connection leads to unnecessary Selects, and a behaviour that in case of failure would make debugging difficult, because all database queries are made in the rendering process and possible sources of error, which are usually in the business logic, can be located badly.

Security issues

- The sharing of data is not encrypted.
- Passwords are not stored as hash.
- There is no explicit reviews of the rights to a particular record.

2. Considerations of a further Development and a Re-launch

2.1 New Requirements and Software Re-launch

Due to the resulting requirements and the aforementioned missing technical specifications and features of the previous software the VITA IT team decided to develop a new application. As far as the design and code base are concerned the application is conceptualised anew. Subsequently, single parts of the old software will be integrated into the new software. This includes the graphical output of data, primarily the generation of SVG graphics and the presentation of the evaluation as a PDF certificate.

Deployment of application logic through CRUD functions as a web

For input and output and data maintenance a web service will be provided. The GUI is here conceived as a thin client that passes the events to the web service and gets answers in the form of records. The service will perform the validity of the data as well as authentication and authorization functions. It is also intended that it also conveys partly the state of the GUI. Thus we achieve:

- A strong decoupling between the GUI and business logic
- A good level of interoperability with other systems
- Easier testability of the application
- A high degree of extensibility and scalability

The communication is effectuated via SOAP messages. The data will be encrypted by means of PKI, this method will also allow a signature of the data or an authorisation and review of the consistency of the data.

2.2 Choice of Programming (Language)

Web services and GUI will be implemented in JAVA. The reasons for this are:

- The preceding project is written in Java, thus some of the classes can be reused.
- Java provides a set of frameworks for creating web services.
- Java is the de facto standard for web services in businesses.
- Web services in Java can be well scaled.

2.3 General Considerations regarding the Design

To develop a first draft of an application, it has been proven well to look first at the data model relating to a user interface. This way it is possible to describe the datasets and records in relations, and class diagrams and data models:

- For the persistence subsequently a ORM framework will be used.
- The proposed top-down web service should be generated from WSDL descriptions. The embedded XML schemes to describe the data model are based on the object-oriented model.
- The WSDL file contains all relevant information to generate top-down the stubs and skeletons. This includes JPA annotations and JAXB-specific bindings (e.g., to use collections for lists). The XSD description of the elements contains all necessary information.

A paradigm shift to a relational description of the model is in both cases not necessary.

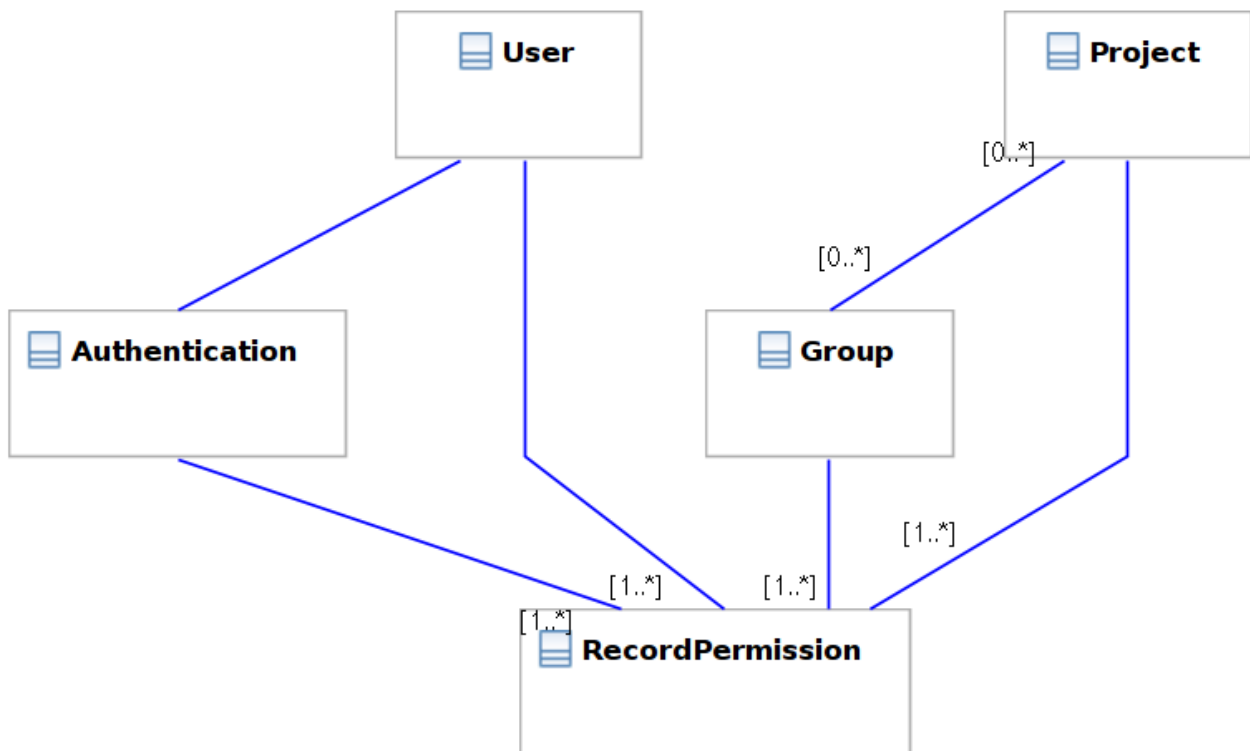
2.4 Simplification of the model to produce a testable prototype

In examining the data structure it can be stated that the basic design can be illustrated with a few classes to wrap up the use cases of the design draft phase. For the sake of clarity and testability in the first phase of the software project, the application logic can be mapped with only a few classes that only require a small number of fields. After the prototype runs stably further classes and fields are added².

To produce a testable prototype the following classes were initially observed that will later represent the user management application:

- User (users of the application)
- Authentication (login information)
- Group (container to compile users)
- Project (containers to combine groups of users for one evaluation context)

Record Permission (adjusts the memberships and access rights to most classes and objects)



In this example, the common minimum relations between the data objects are used: For the description of the associations the same names will be chosen as they are used in the JPA specification.

1 to 1, 1 to n relationships

User (One) "has one" Authentication (One) → OneToOne

User, Authentication ... (One) "has many" Record Permission (Many) → OneToMany

n to m relations

Project (many) "has many" Group (many) → ManyToMany

² Based on these basic use-cases and considerations the different functional areas of the software (user management, project description, inventories of competences, assessment and evidencing/documentation) are being added.

2.5 Prototype of a User Interface

The maintenance of records that require 1 to n relationships is a bit of a problem for user interfaces since the data sets will become complex due to their far-reaching ramifications in hierarchical structures.

One approach (which is also used in file explorers, but also in web applications for navigation in the different levels) would be the representation of a tree structure. Recently, one can observe that more and more the principle of the "breadcrumb trail" prevails in web-applications.

Here the navigation leads to increasingly lower levels of a data structure, wherein the nodes are represented as a path in a bar. This bar can then be used to navigate back to any nodes.

The prototype of the GUI should also follow the principle of the breadcrumb navigation. A data set is selected from a list of data sets. This offers two options:

- the selected record can be edited
- a list of child records can be displayed

Within the list, again a record can be selected, there are again two options.

For each requested record a new tab is created. Using this tab one can navigate to parent records back at any time.



Project Data Inventory Evaluate

Help Logout

User overview First Project Thomas ... Profile IDA - Group

- Application
- Profile
- Project
- Group
- User**

- New**
- Save
- Delete
- Cancel

NAME	DESCRIPTION	↑ ↓
Thomas Grabietz	Informatiker	↑ ↓
Linda Küllmer	Logopädin	
Laila Asleh	Arzthelferin	
Roman Warm	Fachinformatiker	
Tanja Brüb	Optik-Feinmechanikerin	
Tim Scholze	Projektleiter	
Tania Garfias-Macedo	Mathematikerin	

1 zu n: Liste von Benutzern, die durch den angemeldeten Benutzer angelegt wurden

◀◀ ◀ 1 2 3 4 5 ▶ ▶▶

The screenshot displays the LEVEE software interface. At the top left, the LEVEE logo is visible. Below it, a navigation bar contains buttons for 'Project Data', 'Inventory', and 'Evaluate'. To the right of these are 'Help' and 'Logout' buttons. The main content area is divided into several sections. On the left, there are vertical tabs for 'Application', 'Profile', 'Project', 'Group', and 'User'. The 'User' tab is currently selected. In the center, there is a 'User Overview' section with a dropdown menu showing 'Thomas ...' and a description 'Assign to group Authentication'. Below this, there are input fields for 'User' (Thomas Grabietz), 'NAME:' (Thomas Grabietz), and 'BESCHREIBUNG:' (Informatiker). On the right side, there are tabs for 'Profile' and 'IDA - Group'. A red box highlights the 'Thomas ...' dropdown menu with the text 'Reiter: Navigations-Knotenpunkte'. Another red box highlights the 'Informatiker' text in the 'BESCHREIBUNG:' field with the text 'Auswahl eines Benutzers aus der Liste'. At the bottom right, there are four buttons: 'New' (yellow arrow), 'Save' (green checkmark), 'Delete' (red X), and 'Cancel' (purple arrow).

Reiter: Navigations-
Knotenpunkte

m zu n: Zuordnung von
Benutzern zu Gruppen

SELECT	NAME	DESCRIPTION
<input checked="" type="checkbox"/>	Thomas Grabietz	Informatiker
<input type="checkbox"/>	Linda Küllmer	Logopädin
<input type="checkbox"/>	Laila Astleh	Arzthelferin
<input checked="" type="checkbox"/>	Roman Warm	Fachinformatiker
<input checked="" type="checkbox"/>	Tanja Brüb	Optik-Feinmechanikerin
<input type="checkbox"/>	Tim Scholze	Projektleiter
<input type="checkbox"/>	Tania Garfias-Macedo	Mathematikerin

Design of Webservices

Similar to the requirements which arise from the model, and the GUI, CRUD operations for the web service can be described. Through the homogeneity of the data structure and the user interface in the GUI there are 4 basic functions:

retriveList ():

All the records are displayed that the user has at least read access.

- objectId persistObject(object),
- object retriveObject(objectId),
- boolean deleteObject(objectId)

are the respective CRUD data objects and create and update are grouped together for an operation. Depending on whether the id of an object exists or not it is a create operation (no id exists) or update operation (id exists) run.

To return objects, additional information can be sent, which can affect the behaviour of the GUI. These include the rights that a user has on the respective record. If a user does not have write access to a data set, the GUI can disable the Save button.

Besides the return values defined by the function the server can respond with a number of exceptions:

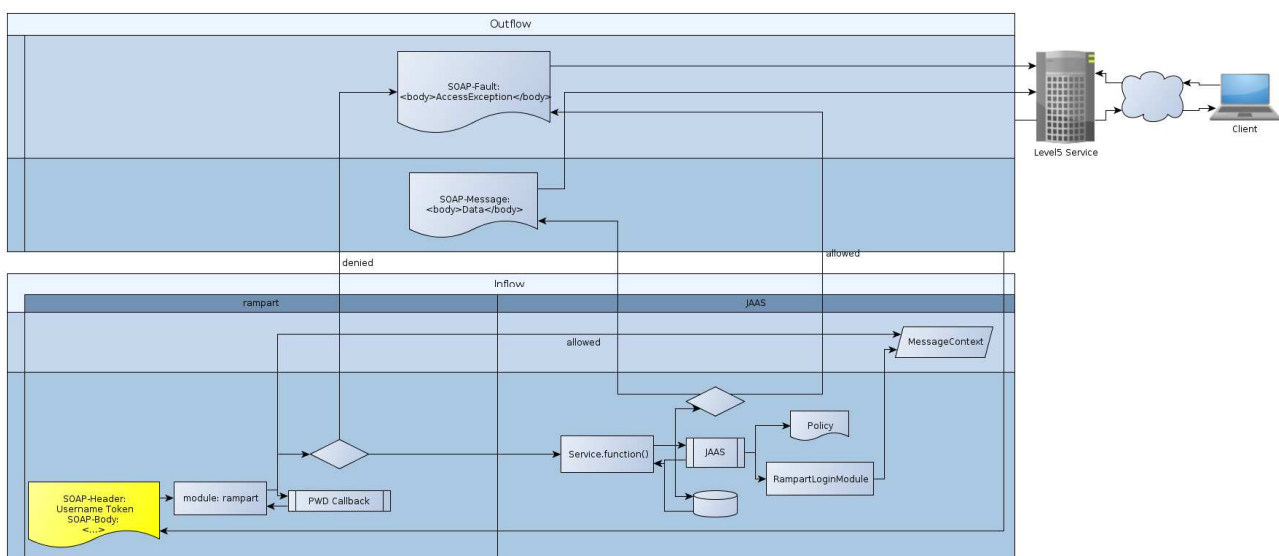
RemoteException: The construction of the connection is faulty.

Fault: The query can not be answered, because an internal (programming) error is present (e.g., null pointer) or the sent item is faulty

[Service name] Fault: Are exceptions that are thrown by the processing logic of the application:

- The described in the XSD types are not valid (e.g. not a number, string too short). The constraints are taken from the WSDL file.
- The user does not have sufficient rights to make operations on the record or the authentication information can not be confirmed.
- The requested record does not exist.

Interleaving of data records is avoided in this design. The data objects have flat structures; references to associated records are only made by their Identifier.



Authentication with Username-Password Token

2.6 Authentication, authorization, and other security goals

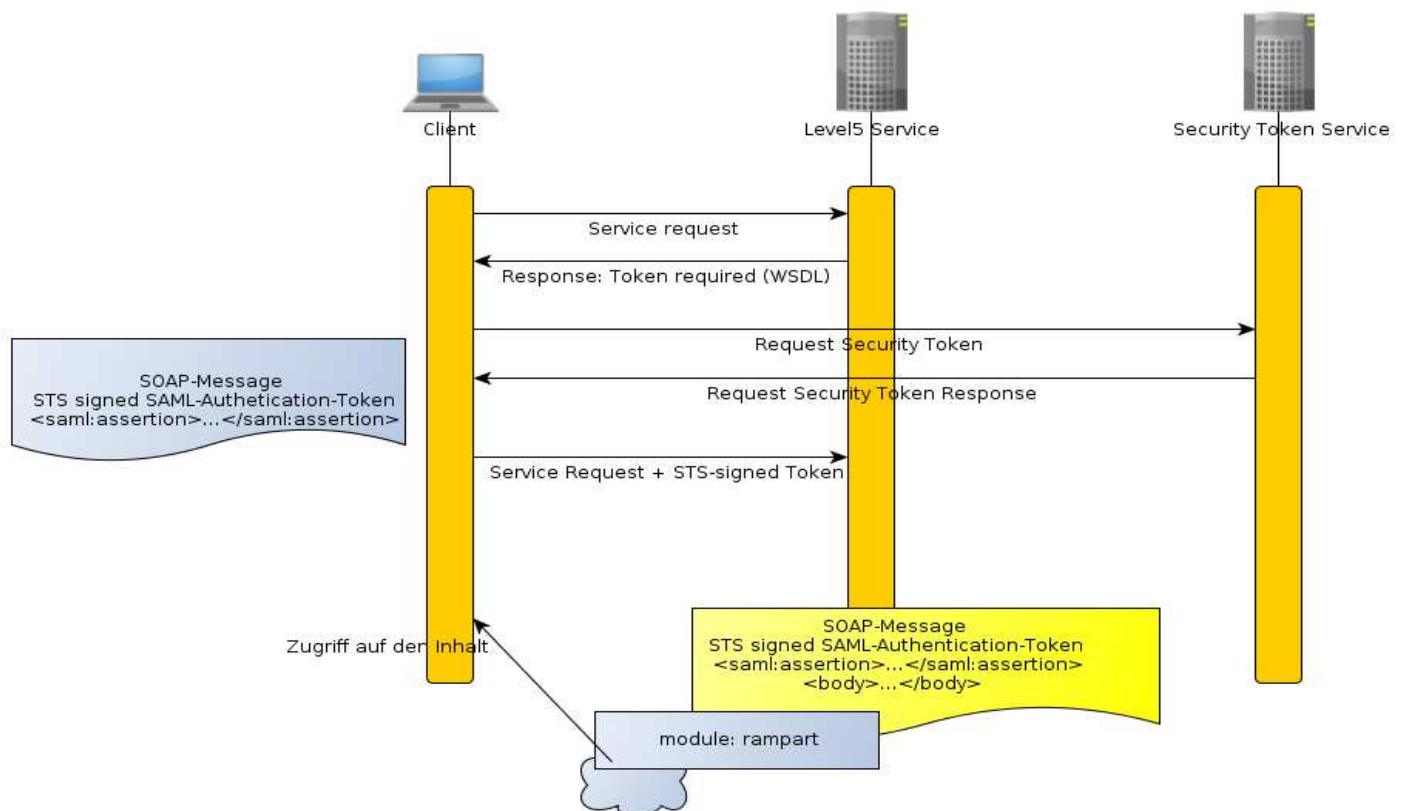
In monolithic applications usually the GUI is closely linked to the business logic. This means that the GUI can be a shield against the application. If data do not appear in the GUI or if buttons and menus are inactive no access to the underlying processing logic is given. The situation is different when the GUI is loosely coupled to a web service. In this case, the web service has to decide whether access can be given to the requested data.

In this application the Rampart framework serves to process authentication information to the web service. Via the JAAS framework it is to be determined subsequently, whether a user is authorised to perform the operation. Rampart is an out-of-the-box solution, which can be set up with a few configuration steps authentication, digital signature and asymmetric encryption. JAAS, however, is primarily a design pattern that defines interfaces that must be implemented depending on the application logic. The Java Security Framework delegates then the verification of the access rights to these interfaces.

2.7 Single Sign On

Security Token Service with SAML-Assertions (Single Sign On)

Function: The client authenticates with a username-password token or a PKI procedures in an STS. If successful, the user receives a SAML assertion, which is then presented to the level5Service . Rampart evaluates the token, and passes the username to the MessageContext, which in turn can be evaluated by JAAS.



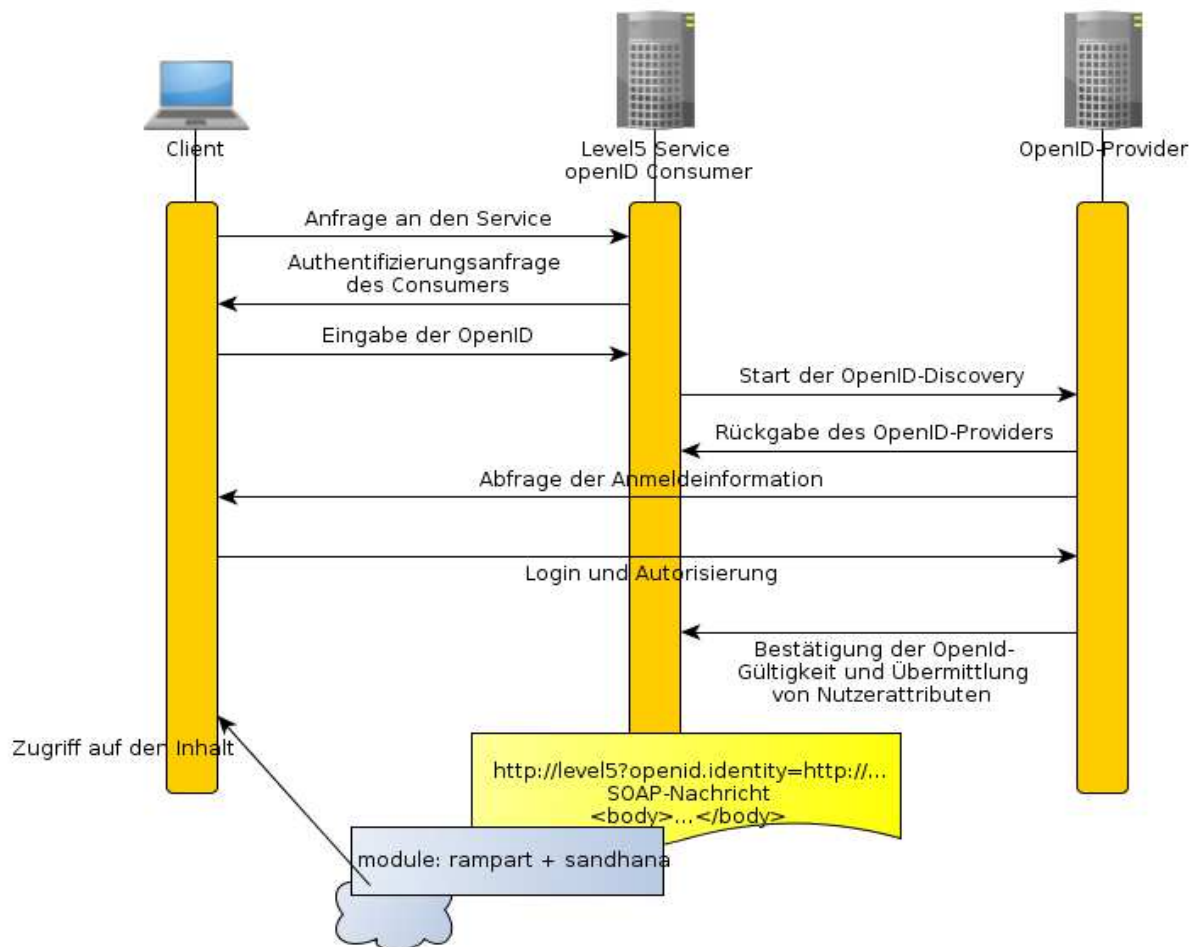
Authentication with SAML + STS

Security Token Service with SAML-Assertions and XACML-Authorisation (Single Sign On)

Besides the SAML assertions there is the possibility to add XACML compliant information. Here the STS could serve both as PEP (Policy Enforcement Point) for formulating the authorising inquiry and as PDP (Policy Decision Point) for authorisation.

In the latter case the JAAS framework of LEVEL5 services would be almost obsolete, as the authentication decision has been already been made.

The strength of XACML SAML + + STS becomes above all obvious, where authentication decisions across systems should be managed. In the context of Level5 finer granulation and standardized user rights would be imaginable.



2.8 Transaction security

Since the communication with the server is operated in “blocking mode”, the server cannot inform on changes to the data.

Thus it is possible that two users who have permissions on an identical record, can make changes at the same time to this. To prevent inadvertent overwriting of data, the server delivers hash values of the current state of the data. A user can overwrite data on the server only if the current hash value of the data coincides with the one that he/she received when reading the data set.

A thin client reference implementation using JEE JSF 2.0:

As user interface a web GUI is provided. This interface is built using JSF Framework version 2.0.

3. Overview of Working Steps

Analysis of the requirements to the software

Requirements

- Future-proof and sustainable, maintenance-friendly architecture is to be ensured
- Interoperability with external systems (e.g. ECVET software), access to interfaces of mobile devices (thin client)
- Loose coupling of the individual modules
- Good testability

Development environment, tools and frameworks

Choice of development tools

- Eclipse JEE, JBoss Tools
- MySQL database (not necessarily, since Hibernate can deal with a number of SQL dialects)
- Hibernate, JPA
- Tomcat
- Network Analysis (soap, TCP Monitor, Wireshark)

Comparison of different Java Web Frameworks

- Axis2
- Metro
- CXF

The instrument of choice: Axis2

Reasons for the decision for the Axis2 Framework

- StaX and Axiom
- Inflow, Outflow: the module chain of Axis2, good expandability
- Ant task based XSTL Templates
- Different scopes, session handling
- WS-Security Rampart
- SOAP & REST support
- Several possible containers

What if? Migration of the Axis2 for JAX-WS specification

Axis2 and Data binding: ADB vs. JAXB

Comparison of Data binding frameworks ADB and JAXB

- Considerations for interoperability
- Data binding and JAX-WS standard
- Tool support, extensions, configurability (notations, inheritance, lists)
- Performance

- Validation
- Readability

Architecture of Services

The various layers:

Persistence layer

- MySql
- Hibernate / JPA
- JAXB, JPA Entities Annotierete
- Generic DAOs as Kommunikationsschicht

The business layer

- The Service Skeleton
- MEP, message receiver
- Service Faults / Exception Handling
- Marshaling Unmarshaling → Validation of the data objects

The presentation layer

- Thin client development with JSF 2.0
- In preparation for HTML CSS, JQuery

Description of Interfaces

- The WSDL file. Xjc bindings and annotations

Framework and library support in the layers

Paradigm shift from the relational layer via object-oriented layer to marshalling

Reasons for the decision of the Hibernate pattern of Detached-Objects

- circular dependencies and recursive, endless graphs
- hibernate proxies and JAXB marshalling
- overhead of nested objects in communication
- failure analysis, debugging

SOAP vs. REST

Reasons for provisionally waiving a REST support

Principally the communication does follow the REST principle

Stateful vs. Stateless

- Jax-WS and sessions?
- Performance, scalability
- Message IDs

Top-down vs. Bottom-up

Consideration of both approaches in the service generation

Top-down and agile development approaches

Is a top-down approach in line with an iterative, agile development?

- Project management, steps, goals
- Iterative development: abstraction of the application, construction of CRUD services from simple to more complex data structures

Top-Down: WSDL, annotations, Ant tasks and xjc bindings

Top-down: data validation: JSR 303 vs. JAXB Validation Event Handler

Since data types are described in XSD validation JSR 303 is unnecessary.

- Adaptation of the Axis2 Message Receiver templates for validation of the data fields

Structure of the message structure, avoiding overhead in communication

On both sides of communication are graphs of objects. To what extent is it possible and useful to synchronise individual objects of the graph via a web service with minimal overhead.

Testability: JUnit framework

- JUnit and testing services in general
- How can test cases be carried out in a data-driven application, so that they are independent of each other?

WS-Security: Security Objectives with Rampart and JAAS

How can the security objectives

- Authenticity
- Confidentiality
- Integrity

be achieved in web services?

The interplay between Rampart and JAAS

Rampart:

- Authentication
- Encryption
- Signing

JAAS

- ACL authorization, access control through capabilities
- Building a dynamic configuration to protect classes and objects, principals and policies with JAAS and Hibernate
- Generalization, encapsulation and integration of access protection in the service methods

The GUI reference implementation, JSF 2.0

Managed Beans and Web stubs

- Implementation of service stubs in the handler layer JSF application
- Forwarding of GUI events to the service
- Caching, updating the data objects
- Exception Handling

Phase Listener for server-side validation of data

To enable the validation of the data by the web service, the JSF's mechanisms (JSR 303, StandardValidators <f:validateX />) have to be invalidated. For this purpose a phase listener is implanted to call at an appropriate point the web service in order to

- valid the data
- forward messages in case of failure to FacesContext.renderResponse () phase and
- store the error messages in the FacesMessage-context.

4. Guided Tour through the software

The software LEVEL5 V 3.0 was released in autumn 2013 after intensive pre-testing. 110 datasets were included.

In September an additional request came up in the framework of the Swiss learning festival in which a simplified and informal “learning ticket” should have been established.

Due to the informal setting of a learning game the layout and the wording of the LEVEL5 learning proofs had to be changed.

Hence the outputs of the process had to be extremely flexible and adaptable since in rather formalised environments (HE, VET) the learners and providers expect “certificates” whereas in rather informal settings other titles like “learning ticket” would have been preferred.

The programmers team from BUPNET and blinc solved this challenge by integrating an open office tool in the process that enables each evaluator to customise his/her own certificates.

One certificate and one learning ticket are attached to this report.

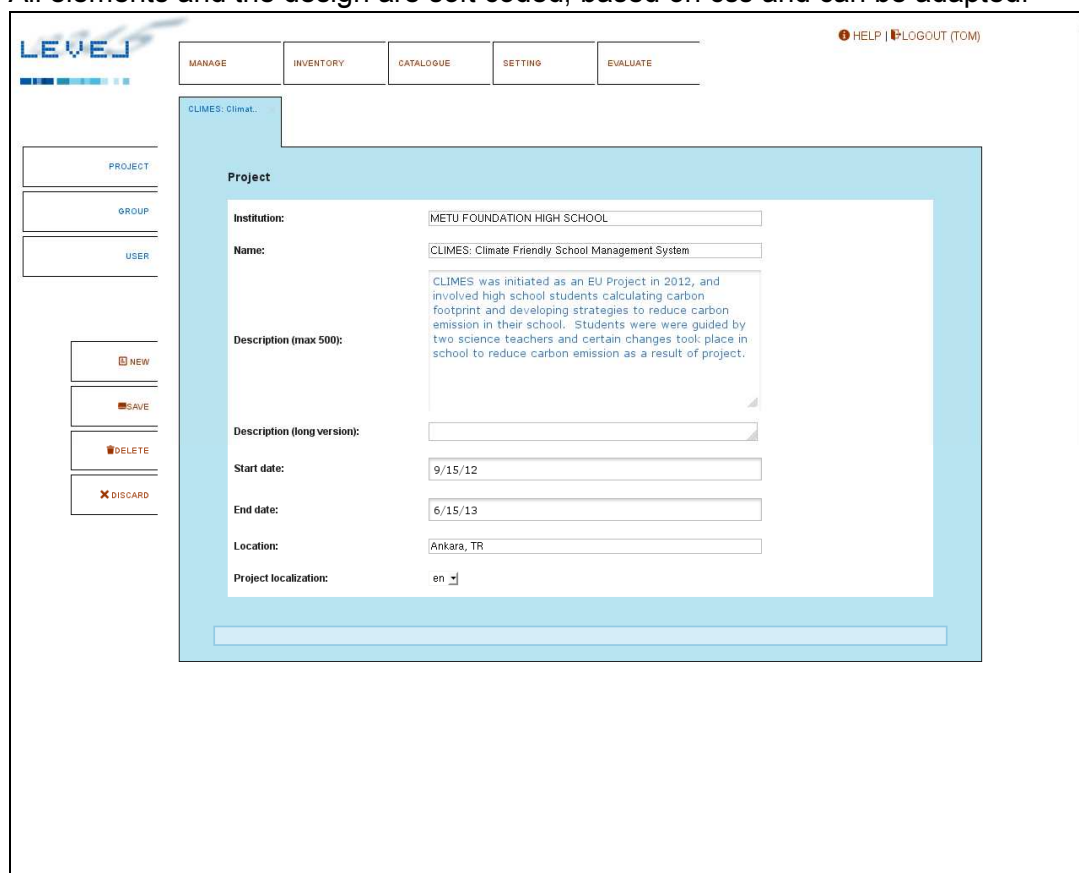
4.1. Step 1: Manage learning project

The following screenshots were taken from the basic design version of the software V 3.0 to illustrate the main functionalities.

The software comes with a global navigation consisting of the main validation steps:

1. manage project
2. Inventory of competences
3. Assessment setting
4. Validation

All elements and the design are soft coded, based on css and can be adapted.



The screenshot displays the 'Project' management interface within the LEVEL5 software. The interface features a top navigation bar with tabs for 'MANAGE', 'INVENTORY', 'CATALOGUE', 'SETTING', and 'EVALUATE'. A sidebar on the left contains buttons for 'PROJECT', 'GROUP', 'USER', and a set of action buttons: 'NEW', 'SAVE', 'DELETE', and 'DISCARD'. The main content area is titled 'CLIMES: Climat.' and contains a form for editing a project. The form fields are as follows:

Institution:	METU FOUNDATION HIGH SCHOOL
Name:	CLIMES: Climate Friendly School Management System
Description (max 500):	CLIMES was initiated as an EU Project in 2012, and involved high school students calculating carbon footprint and developing strategies to reduce carbon emission in their school. Students were guided by two science teachers and certain changes took place in school to reduce carbon emission as a result of project.
Description (long version):	
Start date:	9/15/12
End date:	6/15/13
Location:	Ankara, TR
Project localization:	en

Screenshot 1: Basic learning project interface

The accessibility to those navigation elements is dependent on the user status and the validation scenario and fulfils the requirements:

An evaluator or assessor from a folk-highschool who has to manage hundreds of learners in a high number of learning projects gets a different scenario and permission than the Erasmus student on a self-assessment of only a few competences for one person.

The system comes with a fully fledged learners' management system.

Groups and learners are established in the "manage" area.

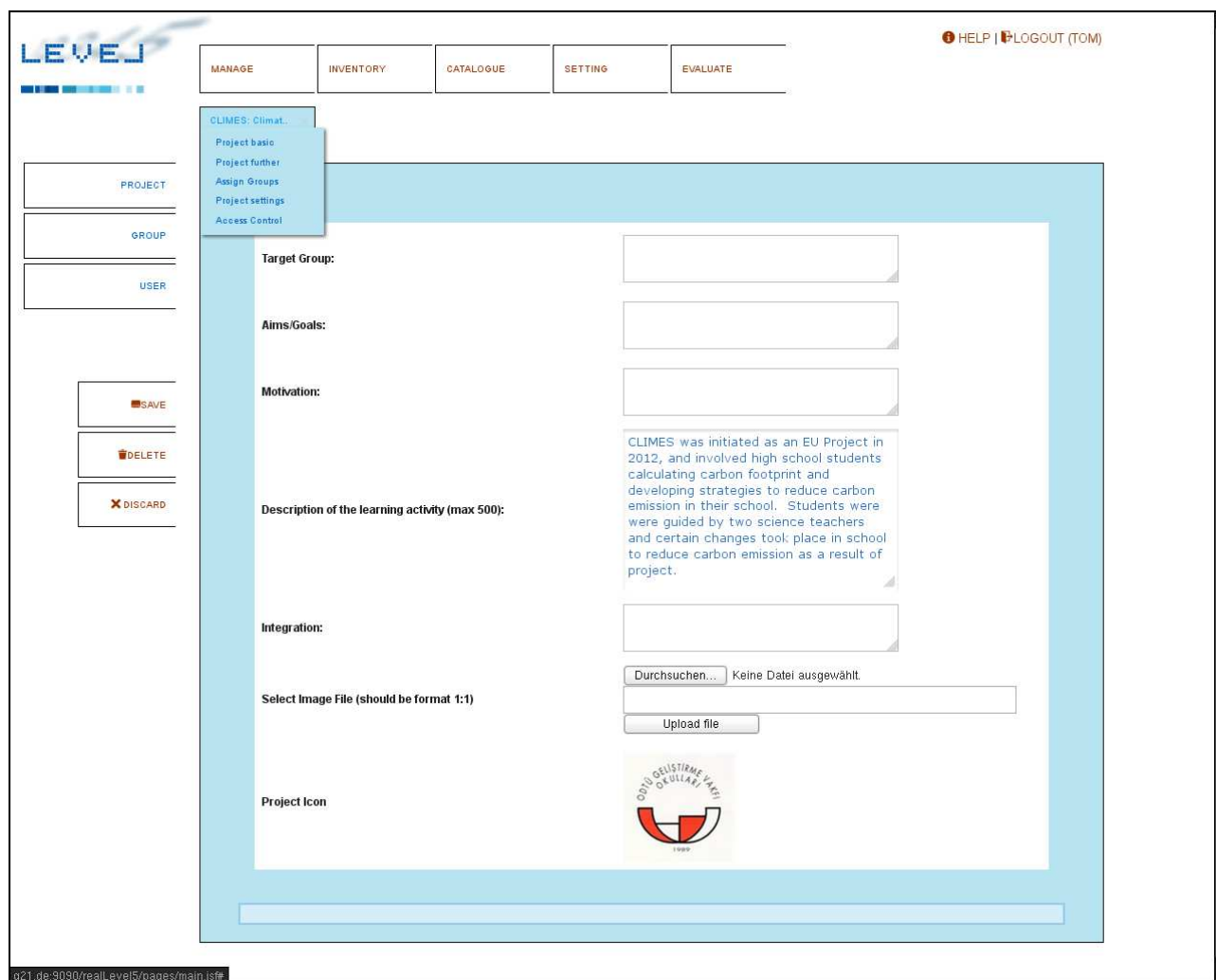
On the left side a set of wizards appears when the functionality is needed, e.g. sub-menus like

- User
- Project
- Group

and editing functions like

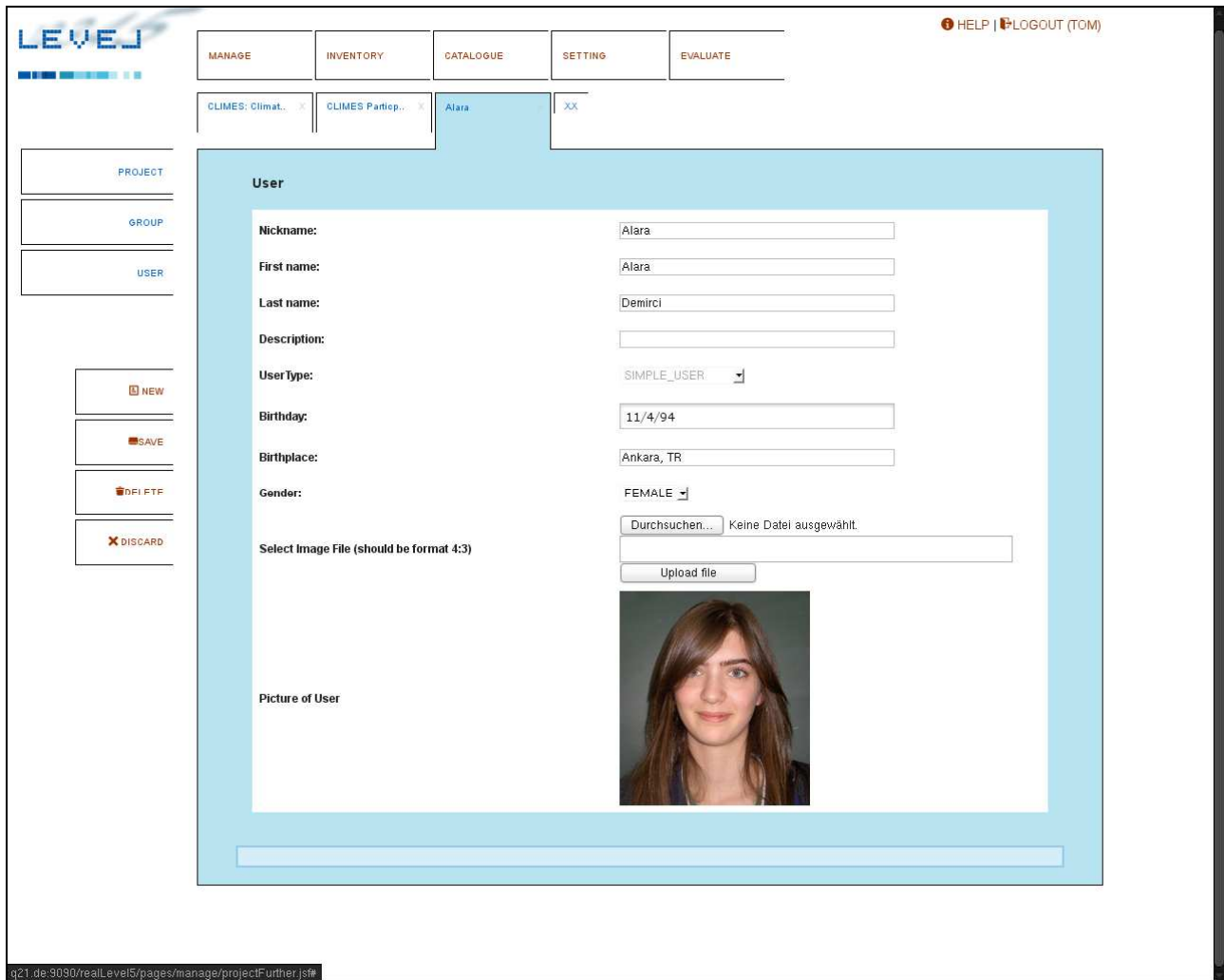
- New
- Edit
- Delete and
- Discard.

The system works with the well known registry system with tabs that can be navigated in a rather intuitive way.



Screenshot 2: Second part of the project description interface

On mouseover of the tabs other sub-menu will appear, context sensitive to the tab.



Screenshot 3: Learners' data interface in the "manage-area"

All relevant data on learning projects, learners' groups and individuals are filled in here.

In LEVEL5 competences are described in terms of knowledge, skills and attitudes. Screenshot 5 shows the user interface in which the user can contextualise these competences by integrating descriptors and indicators for all five levels on the three dimensions.

EQF based systems relate to different dimensions (knowledge, skills and competences) and to eight instead of five quality levels.

The software V 3.0 is able to work with these systems as well as displayed on the next screenshots:

The screenshot displays the LEVEL5 software interface. At the top left is the 'LEVEL5' logo. The top navigation bar includes 'MANAGE', 'INVENTORY', 'CATALOGUE', 'SETTING', and 'EVALUATE'. Below this is a breadcrumb trail: 'CLIMES: Climat.', 'CLIMES Partlep...', 'Alara', 'Willingness to int.', 'new Competence', and 'XX'. On the right, there are links for 'HELP' and 'LOGOUT (TOM)'. The left sidebar contains 'INVENTORY', 'CATEGORY', and 'COMPETENCE' sections, with a vertical menu of 'NEW', 'SAVE', 'DELETE', and 'DISCARD' buttons. The main content area is titled 'Competence' and contains a form with the following fields: 'Name:(id: 5201)' with a value of 'EQF Scale'; 'Description:' with an empty text box; 'Specification:' with a dropdown menu set to 'EQF'; 'Create Reference System:' with a checked checkbox; 'x-Dimension:' with a text box containing 'knowledge'; 'y-Dimension:' with a text box containing 'skills'; 'z-Dimension:' with a text box containing 'competence'; and 'grade count:' with a text box containing '0'. A light blue bar is visible at the bottom of the form area.

Screenshot 6: user interface to create an EQF based reference system

INVENTORY

CATEGORY

COMPETENCE

SAVE

DELETE

DISCARD

knowledge

Level	Name	Scaling	Description (max 250)	Indicators (max 250)
1	Level 1	Basic general knowledge		
2	Level 2	Basic factual knowledge of a field of work or study		
3	Level 3	Knowledge of facts, principles, processes and general concepts, in a field of work or study		
4	Level 4	Factual and theoretical knowledge in broad contexts within a field of work or study		
5	Level 5	Comprehensive, specialised, factual and theoretical knowledge within a field of work or study and an awareness of the boundaries of that		
6	Level 6	Advanced knowledge of a field of work or study, involving a critical understanding of theories and principles		
7	Level 7	Highly specialised		
8	Level 8	Knowledge at the most		

Screenshot 7: user interface to contextualise EQF based reference system

The new software is flexible up to 15 quality grades and different dimension titles.

Hence the main aim, to connect LEVEL5 with EQF systems could be reached already in the software.

4.3. Step 3: Assessment Scenarios

After the description (contextualisation) of the competences in so called reference systems (3 dimensions into 3-15 quality levels each) the next step is the assessment procedure:

Screenshot 8: user interface to include assessment data

Here relevant data about the timing, method and the assessors are included here.

4.4. Step 4: Validation

In the validation (evaluation) section an overview of the assessed group and the assessment status is visible.

Name	Progress: First Assessment Time CLIMES	Progress: Second Assessment Time CLIMES	Evaluate!
Alara			go
Anil			go
Asena			go
Cagla			go
Ezgi			go
Gamze			go
Leyla			go
Sude			go

Screenshot 9: Overview of the validation of one group (here CLIMES students in TR)

	MANAGE	INVENTORY	CATALOGUE	SETTING	EVALUATE
	EQF Scale	CLIMES Compet...	AssessmentSetti...	Evaluation for: A.	XX

BY ASSESSMENT SETTINGS

BY USERS

SAVE

CREATE PDF

CREATE SHORT PDF

Evaluation

Certificate Title:

Certificate Date:

User Name: Anil

Evaluation Summary (max 500):

Category: SOCIAL COMPETENCES

Competence: [Teamwork](#)

Category: PERSONAL COMPETENCES

Competence: [Problem Solving](#)

Name: AssessmentSetting CLIMES

Description: CLIMES

Assigned Competence Set: CLIMES Competence Set

Assigned Project: CLIMES: Climate Friendly School Management System

Assessment Time

First Assessment Time CLIMES: First Assessment Time CLIMES

Time: 9/15/12

external observation

self-rating and assessment in discussion with peers and experts

interview

Second Assessment Time CLIMES: Second Assessment Time CLIMES

Time: 6/15/14

external observation

self-rating and assessment in discussion with peers and experts

interview

Screenshot 10: Validation overview page for one person

Screenshot 10 shows the overview of the assessed competences of a user.

4.5. User Administration

As described the new software is based on different use-cases that create a very complex scenario also in regard to the user administration and the access rights combined with the services that the user shall be able to get.

The software enables the administrators to assign certain scenarios to their learners to facilitate different assessment settings.

According to the two scenarios highlighted above (self-assessment Erasmus student and assessor of a large education institute) different user rights have to be applied.

The screenshot of the administrator interface reveals the high modularity of the system:

The screenshot displays the administrator interface for singular users. The interface is divided into a left sidebar and a main content area. The sidebar contains navigation options: PROJECT, GROUP, and USER. Below these are buttons for NEW, SAVE, DELETE, and DISCARD. The main content area shows a list of users and their permissions, organized into several sections:

- SYSTEM_GENERAL:new User**: A table with 5 columns: Permission, Add, Retrieve, Update, Delete. Rows include PrincipalData:253, PrincipalData:254, PrincipalData:255, User:251, and Authentication:401.
- SYSTEM_OWN:new User**: A table with 5 columns: Permission, Add, Retrieve, Update, Delete. Rows include Evaluation:511, Evaluation:556, and Method:951.
- STANDARD_SHARED:new User**: A table with 5 columns: Permission, Add, Retrieve, Update, Delete.
- SYSTEM_SHARED**: A table with 5 columns: Permission, Add, Retrieve, Update, Delete. Rows include Method:451, Method:453, Method:455, and Method:457.
- SHARED_FACADE:Erasmus Student(251)**: A table with 5 columns: Permission, Add, Retrieve, Update, Delete. Rows include Project:175, Inventory:505, Category:506, Competence:507, and Group:177.

Screenshot 12: administration interface for singular users

With this each user can get individual access rights to:

Read, write, update and delete:

1. learning projects with all functionalities (content, groups, users)
2. inventory of competences or singular competences
3. preformatted or open reference systems, LEVEL5 or EQF-based (or others)
4. assessment systems
5. rating systems and certification tools

With this version 3.0 LEVEL5 comes now with a powerful, highly flexible and modular validation software that covers a broad scope of application scenarios from very simple and easy to use (individual self assessment) up to complex scenarios (validation management purpose).

5. Annexes

5.1. VITA-Certificate of a learner in a Turkish school




Learner-Certificate
CLIMES: Climate Friendly School Management System



Participant
Alara Demirci, 04.11.1994, Female

Additional learning project
CLIMES: Climate Friendly School Management System

Start date: 15.09.2012
End date: 15.08.2014
Institution: METU FOUNDATION HIGH SCHOOL
Location: Ankara, TURKEY




Description of the learning activity
CLIMES was initiated as an EU Project in 2012, and involved high school students calculating carbon footprint and developing strategies to reduce carbon emission in their school. Students were guided by two science teachers and certain changes took place in school to reduce carbon emission as a result of the project.

Evaluated competences
Teamwork, Problem solving

Assessment methods
Self-assessment, teacher assessment, interviews and observations

Evaluator
Meral YILMAZ & Sevl NAZLIER

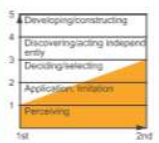



Learning outcome: Teamwork


Competence profile at the beginning	Competence profile at the end
<p>Knowledge : 2 : Understand why teamwork is essential in developing a climate-friendly school management and ways to work in a group toward reaching a goal</p> <p>Skills : 1 : Recognise that measuring carbon footprint of school is essential to initiate teamwork toward enhancing a climate-friendly school management</p> <p>Attitudes : 2 : Show interest in joining a team that aims at promoting a climate-friendly school based on the school's carbon footprint</p>	<p>Knowledge : 3 : Determine how to work in a team and engage in a coordinated work flow by recognizing the skills, qualities and limits of each member</p> <p>Skills : 3 : Contribute actively to teamwork in developing or selecting a climate-friendly action plan to enhance a climate-friendly school</p> <p>Attitudes : 3 : Appreciate variety in team members' skills and decisions; welcome diverse action plans to foster climate-friendly school management</p>




Learning outcome on the learning dimensions

Knowledge	Skills	Attitudes
		
<p>Learned teamwork is essential in this process, and developed skills in engaging in coordinated work flow. Learned team processes and task sharing, directing a discussion, and controlling duties in group works.</p>	<p>Actively contributed to her team to develop and implement an action plan, prepared posters for the school and organized the data to assess carbon emission of school as part of her team.</p>	<p>Became motivated to teamwork, increased awareness of appreciating different perspectives and worked toward common solutions. Attached importance to resourcefulness and creativity in teamwork.</p>

The validation was carried out in the framework of the VITA project funded with support of the European Commission.



The LEVELS® validation system is courtesy of the REVEAL Network for Learning, Validation and Capacity Building



The validation was carried out in the framework of the VITA project funded with support of the European Commission.



The LEVELS® validation system is courtesy of the REVEAL Network for Learning, Validation and Capacity Building



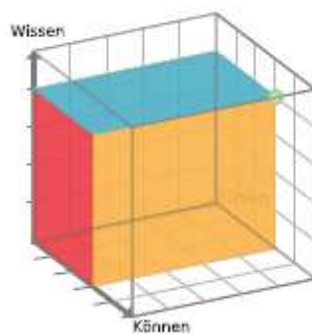
Standard Certificate for a the Turkish VITA school project

5.2. Learning Ticket from the learning festival in Berne (informal)



Persönliches Lern-Ticket vom Lernfestival-Parcours in Bern und Ittigen für
Sonja Schnieders

Der LEVEL5-Würfel zeigt Ihnen die Entwicklung Ihrer Teamwork-Kompetenz in den drei Dimensionen „Wissen“, „Können“ und „Fühlen“:



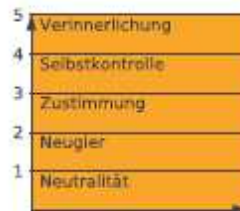
Unten sehen Sie in der ersten Grafik, ob Sie neue Erkenntnisse zum Teamwork gewonnen haben; in der zweiten, was Sie im Team getan und beigetragen haben und in der dritten, inwieweit sich Ihre Gefühle und Einstellungen im Team geändert haben.



Wissen

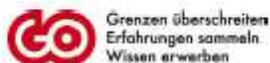


Können



Fühlen

Wenn Sie Interesse an einer detaillierten Darstellung Ihrer Teamwork-Lernergebnisse auf einem LEVEL5-Zertifikat haben, so wenden Sie sich bitte an: service@reveal-eu.org



Customized LEVEL5 certificate issued during the Learning Festival in Berne.