TEAL2.0 SOFTWARE DESIGN

Improving Access to Science and Technology Higher Education in Resource-Poor Institutions through an Open Platform for Technology Enabled Active Learning Environment

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
CONTENTS

Introduction ...........................................................................................................................................2
Features and Components of TEAL2.O .................................................................................................3
TEAL1.O Software Architecture .............................................................................................................9
Domain Model .........................................................................................................................................10
Moodle Integration .................................................................................................................................16
Data Structures .......................................................................................................................................18
Data Model ................................................................................................................................................28
The TEAL2.O Software Design Document translates the software requirements into a representation of the overall architecture, the components, the interfaces and the data necessary in the implementation phase. It spells out more detailed technical specifications and creates a general reference document guiding the next stages of work.

The document should be used in conjunction with the TEAL2.O Analysis model where the requirements are spelled out and the use interface is drafted.
The development of TEAL2.O should start by developing the backbone structure that we have named TEAL1.O. This backbone has five main components, each of which has individual “business” value, i.e. serves a separate delineated function while contributing to overall system performance.

TEAL1.O has the following component structure:

Within the components, there are numerous modules that cannot run as stand-alone functional components but nevertheless allow for decomposition of the development work.

A more detailed graphical representation of the TEAL1.O backbone is presented in the following model:
Here, we present a more detailed view of each component:

**Content Management System**
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
QA and Ranking Engine/ Hosting/ Documentation

The graph below presents the overall system:
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
**DOMAIN MODEL**

- **The Overall System**

  The domain model consists of two major subsystems: the local system and the global system.

- **Local System**

  A local system is a place, physical or virtual, that allows one to construct an academic program and create and run classrooms using tools available in the Global System.

- **Global System**

  The global system has two components:
  - globally accessible cloud-based database that stores all course specifications, learning content, user records (interaction, level of outcome attainment) and user ranking.
  - data-driven Quality Assurance Engine that ranks content based on outcome attainment and usage.

- **Components**

  - **Components of the Local System**

    **Program**

    A program is constructed, for instance, by specifying the broad area of study, the type (UG/PG), the program objectives (POs), number of credits, and a list of courses categorized for example as core, technical electives and non-technical electives. The courses are chosen from a catalogue of courses in the global database. If a desired course is not in the system, it will have to be created and added to the database first. The program should also have a mapping of the course Learning Outcomes (LOs) to the POs. Thus a program is merely a set of specifications. Programs are not saved in a database. They are exclusively user-defined.

    **Program Objectives (POs)**

    The POs characterize the attributes of a program graduate. The objectives are brief, clear statements that describe the desired learning outcomes of instruction; i.e., the specific skills, values, and attitudes students should exhibit at the end of the program or after a certain time period following the completion of the program. These can for instance be specified on the basis of certain accreditation criteria such as those developed by the Washington Accord, the Sydney Accord, or the Seoul Accord.

    **Classroom**

    This is an instance of a course, where the teacher runs a course. In the classroom, teachers interact with students and students interact with content, teachers, and peers. Students also get evaluated in classrooms. Traditional LMSs typically facilitate this process. In TEAL2.0 teachers are expected to...
play the role of mentors, facilitators and evaluators of progress. They may also suggest content in terms of a collection of learning modules. Students may also have the freedom to choose the learning modules in collaboration with the teacher. This requirement is not facilitated in a typical LMS.

Specifically, in a classroom teachers can offer a course that is in the system; students can enrol, interact with learning resources made available in the form of course content modules, interact with the teacher, interact with peers and be assessed. It is also highly desirable if the students can decide – in collaboration with teachers – on the content that best suits them to achieve the LOs of the course. A classroom should thus facilitate the collection of student-specific content modules and map the LOs of the content modules to the course LOs, as well as keep track of student interaction (with the content, teacher, and peers), and student assessments.

**Components of the Global System**

**Course**

Courses are stand-alone components that allow reuse, mixing and modifications. A certain moderation process may be needed. They have no dependency on any other component of the platform and are stored in a global library (global database) that we will refer to as a “course catalogue”.

A course is, much like the program, only a set of specifications. The specifications will include a statement of the broad subject area, the level (PG/UG), the aims and objectives, a benchmark statement, the number of credits, and a list of Learning Outcomes (LOs). The LOs must be mapped to a taxonomy level of choice (such as Blooms or SOLO). It is important to note that a course does not have content such as lecture notes, videos, or assessments.

**Course Content (learning resources) Module**

Course Contents comprise the **core collaborative teaching part of the platform** and are stored in a global library (global database). When a user needs to use a particular content, an instance must be created in a classroom. The instance can be an edited version of the original version. If so, it must be saved as a new version. A certain moderation process and version control are needed.

The content is characterized by a single learning outcome that is mapped to a taxonomy level of choice (for instance Blooms or SOLO). It will also have a credit weight. To encourage modularity, a course content module will be restricted to between 0.5 and 1 credit hours, where a 1-credit hour is defined according to US college standards.

**A course module does have content** such as E-notes, Video Lectures, Demonstrations, Activities/ Assignments, Remote Labs, Virtual Labs, External Resources, etc. This also has assessments that are used to evaluate the student attainment of the specified LO of the content.

If the desired content is not in the system, it will have to be created and added to the database first.

**Course Intended Learning Outcomes (LOs)**
Learning Outcomes are statements that describe significant and essential learning that learners have achieved and can reliably demonstrate at the end of a course or a program. Learning Outcomes identify what the learner will know and be able to do by the end of a course – the essential and enduring knowledge, abilities (skills) and attitudes (values, dispositions) that constitute the integrated learning needed by a graduate of a course or program. The key is that they should be measurable and hence be evaluated.

The learning outcomes approach to education means basing program and curriculum design, content, delivery, and assessment on an analysis of the intended outcome. In this outcomes-based approach to education, the ability to demonstrate learning is the key point.

The structure of a Learning Outcome statement is as follows:

- an action word that identifies the performance to be demonstrated
- a learning statement that specifies what learning will be demonstrated in the performance
- a broad statement of the criterion or standard for acceptable performance.

These outcomes must be mapped to a learning outcome taxonomy of choice such as Bloom's or SOLO.

**Quality Assurance and Ranking Engine**

The functionality of the QA Engine is to rank courses and course content based on their usage and student performance, the actual level of attainment of Learning Outcomes, as well as student feedback. Thus the QA Engine will have to be able to store data on student engagement and on the level of attainment of Learning Outcomes, and to then use a suitable strategy to rank the courses and the course content. The data related to student engagement and student attainment of Learning Outcomes should be anonymously stored in the global database to ensure privacy.

**Standards & Documentation**

Specifications or other precise criteria are designed to be used consistently, as a rule, guideline, or definition to develop a program. They have to focus on what students will need to learn in order to be competitive on the job market.

**Other nomenclature**

- **Assessment** – A student evaluation designed specifically to assess the attainment of a given LO
- **Revision** – Store every version of a course and of course content that is created by contributors. The changes in each version should be easily visible and tractable. A certain moderation process may also be needed
- **Credit** – It is yet to be unambiguously defined what one credit of learning engagement will be.
Roles

Teacher / Classroom Manager
Creates and maintains a classroom.

Student / Learner
Acts as a student/learner in a classroom

Program Creator
Creates a program (local role)

Course Content Contributor
Creates courses and contents (global)

Course Content Moderator
Moderates courses and contents (global).

The domain model is presented graphically below:
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
example, the purchase order for an item can be categorized under pink as it will have the date and time the item was purchased. Therefore, these become tracking details.

- **Yellow: An Active Role.** This category represents an active role that an individual or an organization can play. A person can play a unique role or multiple roles.

- **Green: Party, Place or Thing.** This category can recognize additional attributes like registration number, student's name, etc.

- **Blue: Catalogue like.** In this category, mostly catalogue like descriptions are included. This attribute will have the list of all the details and characteristics for one particular party or activity.
Although Moodle is open source and we can change anything in Moodle to support our needs, the best and most sustainable way to extend it is to write a plugin (sometimes called “a module”). This allows us to incorporate Moodle updates and we will only need to maintain and update the plugins we have developed. It should be reiterated here that the purpose of ensuring Moodle integration is to allow users who already use Moodle as their LMS to incorporate the TEAL2.O features through Moodle in a very convenient way.

Moodle supports a wide range of plugin types. It supports standard and advanced (admin) plugins. The complete list of plugins type details can be found at Plugin types.

Moodle offers a thorough documentation for development. Developer documentation can be found here. It has information on several types of API that are needed to connect with its core and other external systems. These are essential when writing Moodle plugins.

There are several guidelines to follow when developing Moodle plugins. They can be found in dev doc. In Moodle dev doc, they have listed several development tools that are necessary and/or useful. Overview of the communication between Moodle components can be found here.

- **Standard plugins**

  Moodle has a general philosophy of modularity. There are nearly 30 different standard types of plugins and even more sub-plugin types. However, all of these plugin types work in the same way. Blocks and activities are the only small exceptions.

  See Moodle plugins and Moodle sub-plugins for more information.

- **Local plugins**

  The recommended way to add new functionality to Moodle is to create a new standard plugin (module, block, auth, enrol, etc.). The local plugins are mostly suitable for things that do not fit standard plugins.

- **Custom/ local plugins**

  - Local plugins are used in cases when no standard plugin fits, examples are:
    - event consumers communicating with external systems
    - custom definitions of web services and external functions
    - applications that extend Moodle at the system level (hub server, amos server, etc.)
    - new database tables used in core hacks (discouraged)
    - new capability definitions used in core hacks
    - custom admin settings
- extending the navigation block with custom menus.

List of differences from normal plugins:

- always executed last during install/upgrade - guaranteed by order of plugins in `get_plugin_types()`
- are expected to use event handlers - events are intended for communication core-->plugins only, local plugins are the best candidates for event handlers
- can add admin settings to any settings page - loaded last when constructing admin tree
- do not need to have any UI - other plugins are usually visible somewhere.

- **Coding Guidelines**

Developer teams should become thoroughly familiar with Moodle's: coding guidelines. These guidelines should be strictly followed.

- **Tutorial**

There is a Tutorial to help you learn how to write plugins for Moodle from start to finish, while showing you how to navigate the most important developer documentation along the way.

- **Moodle 3.9 Database**

Full list of tables in Moodle database can be found HERE.

Moodle Developer Video Tutorial can be accessed from: YouTube channel. It shows clips with guides and tips for the Moodle development.
### Program Description Tool – Synthesis

**Program**

<table>
<thead>
<tr>
<th>Field/Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Data about the program identification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title of the program</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Acronym of the title</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Country the university that hosts the program is located in</td>
<td>Select from strings – see description column</td>
<td>IT – Italy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IN – India</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No – Norway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ro – Romania</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SL – Sri Lanka</td>
</tr>
<tr>
<td>University that hosts the program</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Study field</td>
<td>Number</td>
<td>According to….. ISCED</td>
</tr>
<tr>
<td>Study cycle (level)</td>
<td>Select from strings</td>
<td>Bachelor/Master/Doctoral</td>
</tr>
<tr>
<td>Attendance</td>
<td>Select from strings</td>
<td>Full time/Part time/Distance learning</td>
</tr>
<tr>
<td>Duration (years)</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Program ID</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Optionally, a Program ID could be generated automatically based on the data above, according to the pattern bellow. The structure of the Program ID could be subject of improvement.

![Diagram](ME RO 01 07 06 01 04)
### Program overview

<table>
<thead>
<tr>
<th>Program objectives</th>
<th>Edit window</th>
<th>Defined according to: Washington accord program, Sydney Accord program, ENAEE European guidelines or others.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intended outcomes</td>
<td>Array of strings</td>
<td></td>
</tr>
<tr>
<td>Competences</td>
<td>Array of strings</td>
<td></td>
</tr>
<tr>
<td>Number of semesters</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Number of weeks/ semester</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Total number of credits (ECTS)</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Supplementary credits (ECTS)</td>
<td>Number</td>
<td>If it applies (e.g. for diploma project / master thesis / …)</td>
</tr>
</tbody>
</table>

### Curriculum

This data section should be organized in array of records. A record stores data about a course.

<table>
<thead>
<tr>
<th>Course Record</th>
<th>Course title</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester</td>
<td>Select from number</td>
<td>1/2</td>
</tr>
</tbody>
</table>
| Course status (content) | Select from strings | Depending on the educational system.  
Example:  
- for the Bachelor level: FC (fundamental course) / DC (course in the study domain)/ SC (speciality course)/ CC (complementary course);  
- for the Master level: PC (proficiency course)/ SC (synthesis course) |
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

## Course Description Tool – Synthesis

<table>
<thead>
<tr>
<th>Course status (attendance type)</th>
<th>Select of strings</th>
<th>CPC (compulsory course) / EC (elective course) / NCPC (non-compulsory course).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture hours / week</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Seminar hours / week</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Laboratory hours / week</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Project hours / week</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Individual study hours (total)</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Number of credits</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Evaluation type</td>
<td>Select of strings</td>
<td>Exam/Colloquium</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field/Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I. Data about the course identification</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of course</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>Select from Strings - see description column</td>
<td>(Bachelor, Master, Doctoral)</td>
</tr>
<tr>
<td>Study year</td>
<td>Select from Number (1-6)</td>
<td></td>
</tr>
<tr>
<td>Semester</td>
<td>Select from Number (1,2)</td>
<td></td>
</tr>
<tr>
<td>Course status (content)</td>
<td>Select from Strings - see description column</td>
<td>- for the Bachelor level: FC (fundamental course) / DC (course in the study domain)/ SC (speciality course) / CC (complementary course); - for the Master level: PC (proficiency course)</td>
</tr>
</tbody>
</table>
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

<table>
<thead>
<tr>
<th>Course status (attendance type)</th>
<th>Select of Strings - see description column</th>
<th>CPC (compulsory course) / EC (elective course) / NCPC (non-compulsory course);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation type</td>
<td>Select of Strings</td>
<td>Exam, Colloquium, aso.</td>
</tr>
</tbody>
</table>

2. Total estimated time and number of credits

<table>
<thead>
<tr>
<th>Number of hours (per week)</th>
<th>Title related to the next 4 fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture hours</td>
<td>Number</td>
</tr>
<tr>
<td>Seminar hours</td>
<td>Number</td>
</tr>
<tr>
<td>Laboratory hours</td>
<td>Number</td>
</tr>
<tr>
<td>Project hours</td>
<td>Number</td>
</tr>
<tr>
<td>Individual study hours (total)</td>
<td>Depending on the educational system. Example: Total independent study hours (study of textbooks, course support, bibliography and notes, additional documentation in libraries, specialized electronic platforms, and field research, preparation of seminars/laboratories/projects, homework, papers, portfolios and essays, examinations, other activities) is used to have sufficient total hours to justify the number of allocated credits.</td>
</tr>
<tr>
<td>Number of credits</td>
<td>Number</td>
</tr>
</tbody>
</table>

3. Prerequisites (if applicable)

<table>
<thead>
<tr>
<th>List of the courses or other requirements</th>
<th>Title related to the next 2 fields</th>
</tr>
</thead>
</table>
### 4. Required resources

<table>
<thead>
<tr>
<th>For course development</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>For seminar/ laboratory/ project development</td>
<td>String</td>
</tr>
</tbody>
</table>

### 5. Intended learning outcomes of the course (ILOs)/Specific competences

<table>
<thead>
<tr>
<th>Competences</th>
<th>Title related to the next 2 fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professional competences</td>
<td>List of selection</td>
</tr>
<tr>
<td>Transversal competences</td>
<td>List of selection</td>
</tr>
</tbody>
</table>

### 6. Course objectives - here we can have Intended learning outcomes of the course (ILOs)

<table>
<thead>
<tr>
<th>General objective</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific objectives</td>
<td>One Text or multiple String fields added on demand (variant/unknown number)</td>
</tr>
</tbody>
</table>

### 7. Topics to be covered/Contents

<p>| Lecture /teaching methods/no. of hours | String or table with inputs | Can be text or a table with columns (Lecture name /teaching methods / no. of hours / observations) with a row for each |</p>
<table>
<thead>
<tr>
<th><strong>Seminar / laboratory / project / teaching-learning methods/ no. of hours</strong></th>
<th><strong>String or table with inputs</strong></th>
<th><strong>Can be text or a table with columns (lesson name / teaching methods / no. of hours / observations) with a row for each seminar/learning unit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bibliography</strong></td>
<td></td>
<td><strong>1 general field or 2 fields separated for lectures and seminars</strong></td>
</tr>
</tbody>
</table>

### 8. Evaluation

<table>
<thead>
<tr>
<th><strong>Lecture evaluation</strong></th>
<th><strong>String or table with inputs</strong></th>
<th><strong>Can be text or a table with columns (Activity type, evaluation criteria, evaluation methods and percentage of the final grade - these need to be addressed even if the field is one big text) Activity type is “Lecture”</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Seminar/ laboratory/ project evaluation</strong></td>
<td><strong>String or table with inputs</strong></td>
<td><strong>Can be text or a table with columns (Activity type, evaluation criteria, evaluation methods and percentage of the final grade - these need to be addressed even if the field is one big text) Activity type is “Seminar/ laboratory/ project”</strong></td>
</tr>
</tbody>
</table>
## Mapping Competencies – Synthesis

<table>
<thead>
<tr>
<th>Professional competencies*</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptors of structural elements’ level of professional competences**</td>
<td>Performing calculations, demonstrations and applications to solve tasks specific to industrial engineering based on knowledge from fundamental sciences.</td>
<td>Associating the knowledge, principles and methods from technical sciences of the field with graphic representations, for solving tasks specific to industrial engineering.</td>
<td>Use of software applications for computer-aided design of products.</td>
<td>Elaboration of manufacturing processes</td>
<td>Design and operation of manufacturing equipment</td>
<td>Planning, management and quality assurance of manufacturing processes.</td>
</tr>
</tbody>
</table>

### KNOWLEDGE

1. Knowledge, understanding of basic concepts, theories and methods of the domain and area of specialization; their proper use in professional communication

<table>
<thead>
<tr>
<th>C1.1</th>
<th>C2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition of important theorems, basic principles and methods specific to fundamental disciplines</td>
<td>Identifying the phenomena, theories, and calculation methods specific to the disciplines in the domain and the spatial design of some bodies or their components</td>
</tr>
</tbody>
</table>

2. Using basic knowledge to explain and interpret various types of concepts, situations, processes.

<table>
<thead>
<tr>
<th>C1.2</th>
<th>C2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performing demonstrations, explaining and interpreting theoretical results in the field to explain and solve problems</td>
<td>Using the knowledge of the disciplines in the field to explain and solve problems</td>
</tr>
</tbody>
</table>

---

* To be stated at most six professional competences
<table>
<thead>
<tr>
<th>SKILLS</th>
<th>C1.3 Applying general rules for specific problems of engineering sciences</th>
<th>C2.3 Design of simple parts and subassemblies using appropriately the standards and norms in force</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Application of basic principles and methods for solving well-defined problems / situations, typical for the domain in conditions of qualified assistance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1.4 Solving problems of medium complexity and interpreting their results</td>
<td>C2.4 Design of subassemblies and assemblies of medium complexity using appropriately the standards and norms in force</td>
<td></td>
</tr>
<tr>
<td>4. Appropriate use of standard evaluation criteria and methods to assess the quality, merits and limitations of processes, programs, projects, concepts, methods and theories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1.5 Choosing the optimal method</td>
<td>C2.5 Achieving projects of</td>
<td></td>
</tr>
<tr>
<td>5. Elaboration of professional projects with projects, etc. associated with the domain</td>
<td>use or explanation of theorems or phenomena associated with engineering sciences and interpret theoretical or experimental results</td>
<td></td>
</tr>
</tbody>
</table>

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
| Minimum performance standards for competency assessment: | Standard Identify and explain the basic concepts, principles and methods in the fundamental disciplines. Minimum level: Correctly solving the calculations and mathematical / physical problems of medium complexity, specific to engineering sciences. | Standard Identify and explain the concepts, principles, phenomena, parameters and methods of basic engineering sciences. Minimum level: Correctly solving the of problems of medium complexity, in the domain of industrial engineering sciences, including by using the computing and CAD software | Minimum performance standards for competency assessment |

<table>
<thead>
<tr>
<th>Level descriptors of cross competences **</th>
<th>Cross competences</th>
<th>Minimum performance standards for competency assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Responsible execution of professional</td>
<td>-----------------</td>
<td>------------------------</td>
</tr>
</tbody>
</table>

---

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
<table>
<thead>
<tr>
<th>tasks, in conditions of limited autonomy and qualified assistance</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Familiarization with the roles and activities specific to teamwork and distribution of tasks for subordinate level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Awareness of the need for continuous training; efficient use of learning resources and techniques for personal and professional development</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Database Structure

1. **PROGRAM_CONTRIBUTOR**
   - PROGRAM_CONTRIBUTOR_ID <VARCHAR2> (Unique ID - Primary Key)
   - PROGRAM_CONTRIBUTOR_NAME <VARCHAR2>
   - PROGRAM_CONTRIBUTOR_INSTITUTION <VARCHAR2>
   - PROGRAM_CONTRIBUTOR_SYSTEM <VARCHAR2>
   - PROGRAM_CONTRIBUTOR_EMAIL <VARCHAR2>

2. **STUDENT**
   - STUDENT_ID <VARCHAR2> (Unique ID - Primary Key)
   - STUDENT_NAME <VARCHAR2>
   - STUDENT_PROGRAM_TEMPLATE_ID <VARCHAR2> (Foreign Key from PROGRAM TEMPLATE)
   - STUDENT_REG_NO <VARCHAR2> (Unique)
   - STUDENT_EMAIL <VARCHAR2>
   - STUDENT_SPECIALIZATION <VARCHAR2> (Optional)
   - STUDENT_INSTITUTION <VARCHAR2> (Optional)
   - STUDENT_SYSTEM <VARCHAR2> (Optional)

3. **TEACHER**
   - TEACHER_ID <VARCHAR2> (Unique ID - Primary Key)
   - TEACHER_NAME <VARCHAR2>
   - TEACHER_EMAIL <VARCHAR2>
   - TEACHER_INSTITUTION <VARCHAR2>
   - TEACHER_SYSTEM <VARCHAR2>
   - TEACHER_SPECIALIZATION <VARCHAR2>
   - TEACHER_DEPARTMENT <VARCHAR2>

4. **PROGRAM_MANAGER**
   - PROGRAM_MANAGER_ID <VARCHAR2> (Unique ID - Primary Key)
   - PROGRAM_MANAGER_NAME <VARCHAR2>
   - PROGRAM_MANAGER_INSTITUTION <VARCHAR2>
   - PROGRAM_MANAGER_SYSTEM <VARCHAR2>
   - PROGRAM_MANAGER_EMAIL <VARCHAR2>

5. **COURSE_CATALOG**
   - COURSE_CATALOG_ID <VARCHAR2> (Unique ID - Primary Key)
6. PROGRAM_TEMPLATE
   - PROGRAM_TEMPLATE_ID <VARCHAR2> (Unique ID - Primary Key)
   - PROGRAM_TEMPLATE_PROGRAM_CONTRIBUTOR_ID <VARCHAR2> (Foreign Key to PROGRAM_CONTRIBUTOR)
   - PROGRAM_TEMPLATE_DISCIPLINE <VARCHAR2>
   - PROGRAM_TEMPLATE_OUTCOMES <VARCHAR2> (Foreign Key to the PROGRAM_OUTCOME)
   - PROGRAM_TEMPLATE_STAND_DOC <VARCHAR2> (Foreign Key to the STANDARD_DOCUMENTATION)
   - PROGRAM_TEMPLATE_RULES_REGULATIONS <VARCHAR2> (Foreign Key to the RULES_REGULATIONS)
   - PROGRAM_TEMPLATE_RANKINGS <VARCHAR2>
   - PROGRAM_TEMPLATE_TITLE <VARCHAR2>
   - PROGRAM_TEMPLATE_AWARD <VARCHAR2>
   - PROGRAM_TEMPLATE_ENTITY_REQ <VARCHAR2>
   - PROGRAM_TEMPLATE_CAREER_PROSPECTS <VARCHAR2>
   - PROGRAM_TEMPLATE_EXTENT <VARCHAR2>
   - PROGRAM_TEMPLATE_TOTAL_CREDITS <VARCHAR2>

7. PROGRAM_MANAGER
   - PROGRAM_MANAGER_ID <VARCHAR2> (Unique ID - Primary Key)
   - PROGRAM_MANAGER_NAME <VARCHAR2>
   - PROGRAM_MANAGER_INSTITUTION <VARCHAR2>
   - PROGRAM_MANAGER_SYSTEM <VARCHAR2>
   - PROGRAM_MANAGER_EMAIL <VARCHAR2>

8. SCHEDULE
   - SCHEDULE_ID <VARCHAR2> (Unique ID - Primary Key)
   - SCHEDULE_NAME <VARCHAR2>
   - SCHEDULE_PERIOD <VARCHAR2>
   - SCHEDULE_CONTENT_CODE <VARCHAR2>

9. CLASSROOM_PLAN
   - CLASSROOM_PLAN_ID <VARCHAR2> (Unique ID - Primary Key)
   - CLASSROOM_PLAN_TOPIC <VARCHAR2>
VALIDATION OF PLANNED REQUIREMENTS

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

10. CLASSROOM_SCHEDULE
   - CLASSROOM_SCHEDULE_ID <VARCHAR2> (Unique ID - Primary Key)
   - CLASSROOM_SCHEDULE_ACTIVITY <VARCHAR2>
   - CLASSROOM_SCHEDULE_CONTENT_CODE <VARCHAR2>
   - CLASSROOM_SCHEDULE_DETAILS <VARCHAR2>
   - CLASSROOM_SCHEDULE_INSTRUCTORS <VARCHAR2>
   - CLASSROOM_SCHEDULE_SCHEDULE <VARCHAR2> (Foreign Key to SCHEDULE)
   - CLASSROOM_SCHEDULE_SCHEDULE_ORDER <VARCHAR2>

11. CLASSROOM
    - CLASSROOM_ID <VARCHAR2> (Unique ID - Primary Key)
    - CLASSROOM_NAME <VARCHAR2>
    - CLASSROOM_PLAN_ID <VARCHAR2> (Foreign Key to CLASSROOM_PLAN)
    - CLASSROOM_DISCIPLINE_ID <VARCHAR2> (Foreign Key to DISCIPLINE)
    - CLASSROOM_AUTHOR_TEACHER_ID <VARCHAR2> (Foreign Key to TEACHER)
    - CLASSROOM_REG_STUDENTS <VARCHAR2> (Foreign Key to STUDENT)
    - CLASSROOM_SCHEDULE <VARCHAR2> (Foreign Key to CLASSROOM_SCHEDULE)
    - CLASSROOM_COURSE_REQ <VARCHAR2> (Foreign Key to COURSE)
    - CLASSROOM_ILO <VARCHAR2> (Foreign Key to ILO)
    - CLASSROOM_SUGG_CONT <VARCHAR2>
    - CLASSROOM_REMARKS <VARCHAR2>

12. STANDARD_DOCUMENTATION
    - STAND_DOC_ID <VARCHAR2> (Unique ID - Primary Key)
    - STAND_DOC_STANDARDS <VARCHAR2>
    - STAND_DOC_DESCRIPTION <VARCHAR2>

13. PROGRAM_OUTCOME
    - PROGRAM_OUTCOME_ID <VARCHAR2> (Unique ID - Primary Key)
    - PROGRAM_OUTCOME_CODE <VARCHAR2>
    - PROGRAM_OUTCOME_DOMAIN <VARCHAR2>
    - PROGRAM_OUTCOME_INDICATOR <VARCHAR2>
14. RULES_REGULATIONS
   o RULES_REGULATIONS_ID <VARCHAR2> (Unique ID - Primary Key)
   o RULES_REGULATIONS_CODE <VARCHAR2>
   o RULES_REGULATIONS_TYPE <VARCHAR2>
   o RULES_REGULATIONS_DESCRIPTION <VARCHAR2>

15. COURSE_CONTRIBUTOR
   o CONTRIBUTOR_ID <VARCHAR2> (Unique ID - Primary Key)
   o CONTRIBUTOR_NAME <VARCHAR2>
   o CONTRIBUTOR_EMAIL <VARCHAR2>
   o CONTRIBUTOR_SPECIALIZATION <VARCHAR2>
   o CONTRIBUTOR_INSTITUTION <VARCHAR2>

16. COURSE_MODERATOR
   o MODERATOR_ID <VARCHAR2> (Unique ID - Primary Key)
   o MODERATOR_CONTRIBUTOR_ID <VARCHAR2> (Foreign Key to COURSE_CONTRIBUTOR)
   o MODERATOR_NAME <VARCHAR2>
   o MODERATOR_EMAIL <VARCHAR2>
   o MODERATOR_SPECIALIZATION <VARCHAR2>

17. DISCIPLINE
   o DISCIPLINE_ID <VARCHAR2> (Unique ID - Primary Key)
   o DISCIPLINE_NAME <VARCHAR2>
   o DISCIPLINE_FIELD <VARCHAR2>
   o DISCIPLINE_SUB-DISCIPLINE <VARCHAR2>
   o DISCIPLINE_DESCRIPTION <VARCHAR2>

18. CLASSIFICATION
   o CLASSIFICATION_ID <VARCHAR2> (Unique ID - Primary Key)
   o CLASSIFICATION_DISCIPLINE_ID <VARCHAR2> (Foreign Key to DISCIPLINE)
   o CLASSIFICATION_SUBJECTAREA <VARCHAR2>
   o CLASSIFICATION_UG/PG <VARCHAR2>
   o CLASSIFICATION_LEVEL <NUMERIC>

19. SUBJECT_AREA
   o SUBJECT_AREA_ID <VARCHAR2> (Unique ID - Primary Key)
   o SUBJECT_AREA_CLASSIFICATION_ID <VARCHAR2> (Foreign Key to CLASSIFICATION)
   o SUBJECT_MAJOR <VARCHAR2>
   o SUBJECT_FIELD <VARCHAR2>
   o SUBJECT_STUDY <VARCHAR2>
20. RANKING
   o RANKING_ID <VARCHAR2> (Unique ID - Primary Key)
   o RANKING_USAGE <VARCHAR2>
   o RANKING_FEEDBACK <VARCHAR2>
   o RANKING_RANKING <VARCHAR2>

21. COURSE
   o COURSE_ID <VARCHAR2> (Unique ID - Primary Key)
   o COURSE_NAME <VARCHAR2>
   o COURSE_CREDITS_ID <VARCHAR2> (Foreign Key to CREDIT)
   o COURSE_ILO <NUMERIC> (Foreign Key to ILO)
   o COURSE_AUTHOR <VARCHAR2>
   o COURSE_DISCIPLINE <VARCHAR2> (Foreign Key to DISCIPLINE)
   o COURSE_PREREQUISIT <VARCHAR2>
   o COURSE_RANKING <VARCHAR2>
   o COURSE_BENCHMARK <VARCHAR2> (Foreign Key to BENCHMARK)

22. CONTENT_MODULE
   o CONTENT_ID <VARCHAR2> (Unique ID - Primary Key)
   o CONTENT_CREDIT <NUMERIC>
   o CONTENT_ILO <VARCHAR2> (Foreign Key to ILO)
   o CONTENT_CONTRIBUTOR_ID <VARCHAR2> (Foreign Key to CONTENT_CONTRIBUTOR)
   o CONTENT.MODERTOR_ID <VARCHAR2> (Foreign Key to CONTENT.MODERATOR)
   o CONTENT.DESCIELINE <VARCHAR2> (Foreign Key to DISCIPLINE)
   o CONTENT.CONTENT <VARCHAR2> (Foreign Key to CONTENT)
   o CONTENT.ASSESSMENT <VARCHAR2> (Foreign Key to ASSESSMENT)
   o CONTENT.AUTHOR <VARCHAR2>
   o CONTENT.PARENTCONTENTMODULE <VARCHAR2>
   o CONTENT.VERSION <VARCHAR2>
   o CONTENT.REVISIONHISTOY <VARCHAR2> (Foreign Key to REVISION_HISTORY)
   o CONTENT.BENCHMARK <VARCHAR2> (Foreign Key to BENCHMARK)
   o CONTENT.STUDENT_FEEDBACK <VARCHAR2> (Foreign Key to STUDENT_FEEDBACK)
   o CONTENT_RANKING <NUMERIC> (Foreign Key to RANKING)

23. CREDIT
   o CREDIT_ID <VARCHAR2> (Unique ID - Primary Key)
   o CREDIT_HOURSOFSPENT <VARCHAR2>
   o CREDIT.CREDIT <VARCHAR2>
24. STUDENT_FEEDBACK
   - FEEDBACK_ID <VARCHAR2> (Unique ID - Primary Key)
   - FEEDBACK_TITLE <VARCHAR2>
   - FEEDBACK_RATING <VARCHAR2>
   - FEEDBACK_DESCRIPTION <VARCHAR2>

25. REVISION_HISTORY
   - REVISIONH_ID <VARCHAR2> (Unique ID - Primary Key)
   - REVISIONH_CONTENT_MODULE <VARCHAR2>
   - REVISIONH_REVISION <VARCHAR2> (Foreign Key to REVISION)

26. REVISION
   - REVISION_ID <VARCHAR2> (Unique ID - Primary Key)
   - REVISION_NUMBER <NUMERIC>
   - REVISION_REVIEWER <VARCHAR2>
   - REVISION_DATATIME <DATE>

27. ILO
   - ILO_ID <VARCHAR2> (Unique ID - Primary Key)
   - ILO_OUTCOME <VARCHAR2>
   - ILO_ILOTYPE <VARCHAR2>
   - ILO_DESCRIPTION <VARCHAR2>

28. CONTENT
   - CONTENT_ID <VARCHAR2> (Unique ID - Primary Key)
   - CONTENT_AUTHOR <VARCHAR2>
   - CONTENT_TITLE <VARCHAR2>
   - CONTENT_DESCRIPTION <VARCHAR2>

29. ASSESSMENT
   - ASSESSMENT_ID <VARCHAR2> (Unique ID - Primary Key)
   - ASSESSMENT_AUTHOR <VARCHAR2>
   - ASSESSMENT_TITLE <VARCHAR2>
   - ASSESSMENT_DESCRIPTION <VARCHAR2>

30. CONTENT_CONTRIBUTOR
   - CONTRIBUTOR_ID <VARCHAR2> (Unique ID - Primary Key)
   - CONTRIBUTOR_NAME <VARCHAR2>
   - CONTRIBUTOR_EMAIL <VARCHAR2>
   - CONTRIBUTOR_INSTITUTION <VARCHAR2>
   - CONTRIBUTOR_SPECIALIZATION <VARCHAR2>
31. **CONTENT.MODERATOR**
   - MODERATOR_ID <VARCHAR2> (Unique ID - Primary Key)
   - MODERATOR_NAME<VARCHAR2>
   - MODERATOR_EMAIL<VARCHAR2>

32. **ILO_TYPE**
   - ILOT_ID <VARCHAR2> (Unique ID - Primary Key)
   - ILOT_ILO <VARCHAR2> (Foreign Key to ILO)
   - ILOT_CODE <NUMERIC>
   - ILOT_TITLE<VARCHAR2>
   - ILOT_TAXONOMY<VARCHAR2>
   - ILOT_KEYWORDS<VARCHAR2>
   - ILOT_DESCRIPTION<VARCHAR2>

33. **USER**
   - USER_ID <VARCHAR2> (Unique ID - Primary key)
   - USER_NAME<VARCHAR2>
   - USER_TYPE <VARCHAR2>
   - USER_INSTITUTION <VARCHAR2>
   - USER_EMAILID <VARCHAR2>

34. **PROGRAM_OBJECTIVE**
   - PROGRAM_ID <VARCHAR2> (Foreign Key)
   - PROGRAM_OBJ_DESC <VARCHAR2>
   - PROGRAM_ACCRED <VARCHAR2> (Washington Accord, the Sydney Accord, or the Seoul Accord)