Software engineering project LINGI2255

Case specification

Table of contents

1   Introduction

This document presents the expectations of Eureduka, the nonprofit partner of the software engineering project LINGI2255, for the development of the Oscar educational website.

The Oscar landing page :

https://www.oscar.education

UCL references :

https://uclouvain.be/cours-2017-lingi2255

https://lingi2255project.slack.com/

https://moodleucl.uclouvain.be/course/view.php?id=7599

1.1   Background of this project

Several studies attest to the inequity of the Belgian education system. More than anywhere else in Europe, pupils suffer from grade repetition (which everyone agrees is generally counterproductive) : 47 % of students at age 15 have repeated a grade at least once in French-speaking Belgium. Many students who have repeated a grade or two are relegated to vocational education. Sometimes, students drop out of school before they even reach the end of secondary school. If we want the school to fulfil its mission of social emancipation, we need to offer much more support to disadvantaged pupils as soon as learning difficulties appear. Ongoing monitoring of students' progress enables them to be taught in the most appropriate way. It paves the way for differentiated instruction for students of different levels in the same class. This ongoing monitoring is an arduous task. Laurent Fourny used to be a high school mathematics teacher until the creation of the not-for-profit organisation Eureduka. He designed a computerized tool that would facilitate the assessment of students, and then provide them with educational resources adapted to their needs.

1.2   The nonprofit organisation Eureduka

The nonprofit organisation Eureduka was founded in August 2016 to participate in the improvement of the learning/teaching experience in Belgium. Eureduka supports the Oscar project, which is her only activity at the moment.

The Eureduka team is made up as follows:

- Luc Cooremans, general coordinator ;

- Laurent Fourny, managing director ;

- Justine Lottin, graphics designer ;

- Emilie Sickinghe, pedagogical coordinator.

Maher Chemseddine is in charge of IT development *ad interim* (until July 2018). Robin Descamps, a master 2 student in computer science, will coordinate the *LINGI2255 software engineering project* on behalf of Eureduka. We have made sure to document

the code to help you understand it. Robin Descamps will answer - subject to his availability - any question about the code that has not been answered in this documentation. Robin's email : robin.descamps@student.uclouvain.be. He can also answer your questions through Slack.

1.3 The Oscar website

1.3.1 Description

The Oscar website site aims to provide elementary and secondary school teachers with support for teaching mathematics. It facilitates student assessment and provides the teacher with a skill overview for each student. It also helps students in their learning process. Finally, it introduces visitors to the functions of the site and the Eureduka team.

Student assessment: Oscar helps organise a diagnosis at the beginning of the year and tests throughout the year.

Skill overview: Oscar presents each student's cognitive situation in the form of a set of skills, each skill being "mastered","unmastered" or "untested". The status of a skill is updated after an assessment, or modified directly by the teacher. At this stage, Oscar prompts the student with one context (usually with a couple of related questions) to test a single skill.

Learning support (under teacher supervision): Oscar provides students with access to a variety of educational resources (videos, animations, games, PDF files, etc.) that meet their specific needs.

Presentation of the functions of the site and the team: building trust, convincing people of relevance (objective) and usefulness (subjective), making them want to register.

We hope that the site will be useful and enjoyable for teachers and students alike. We have a communication and promotion budget that will facilitate its dissemination to this target audience. Among other promotional initiatives, we will have a booth at the Éduc' show in Charleroi in October 2017.

1.3.2 Development

The Oscar website is being actively promoted during the school year 2017-2018 with the aim of reaching 30,000 users (teachers and students alike) by the beginning of the 2018 school year. We have several audiences for whom the user experience (UX) needs to be adapted:
- primary school teachers, who have generalist training. Oscar's basic functions can only help them in the mathematics course. We believe that only female teachers in grades 4-6 of primary school will use Oscar;
- Mathematics teachers in secondary education, who have specific training in mathematics didactics.
- pupils in grades 4 to 6 of primary school, as it seems premature to us to encourage younger pupils to use computer tools such as Oscar. The youngest users would therefore be about 9 years old, which imposes particular constraints

on security and privacy policy. We estimate that they will connect mainly by computer or tablet, both at school and at home.

- secondary school students, who (almost all) have an intensive use of smartphones. They will connect mainly by smartphone at home or on the move and by computer at school (where the use of smartphones in class is still largely forbidden).

The high number of users, their various profiles and the young age of some of them impose particular constraints, as described in the section 2.2.

1.4 Purpose of the software engineering project LINGI2255

We want to extend the service offered to teachers and students, building on these first functions. We ask you to develop new features for the Oscar website. The software engineering project is not limited to obtaining a functional module; it aims to integrate your piece of software into the Oscar website. We ask you to keep as much as possible the structure of the database. Any discrepancies should be discussed with Robin Descamps.

Your work - if it meets expectations - will improve the learning experience of many students in Belgium. It will gain wide visibility, which is an additional argument for demonstrating your programming skills to a potential employer.

2 IT development

2.1 Current state

The attached technical documentation describes the current state of the software and the coding rules for further development. A lot of information is also available within the code.

2.2 Technical constraints

We want the produced code to be of high quality and satisfy the constraints we apply to our internal IT development. We consider that the quality of the website is based on the following indicators (based on ISO 9126 norm):

- Interoperability.

- Security.

- User-friendliness : ease of use, readability, etc.

- Performance : this criterion mainly concerns the response time, the availability of the application and the scalability (in case of intensive use).

- Maintainability : the code must comply with the standards set up with clear documentation.

3 Functions to be developed

Here below, we describe the functions to be developed, by category (skill assessment, work organisation, etc.). Each group has to choose three functions and submit them through the online survey (on Moodle) by descending order of preference. Each group will subsequently get assigned one of the chosen functions as their main task for this project.

Function	Section	Description
A	3.1.1	Questions with adaptive answer boxes
B	3.1.2	Questions aiming to assess drawing skills
C	3.1.3	Questions aiming to assess statistical graphics skills
D	3.1.4	Evaluation of algebraic expressions
E	3.1.5	Automated question/answer creation (one group or two groups)
F	3.1.6	Scan and import of handwritten answers
G	3.1.7	Questionnaire/survey towards students or teachers
H	3.2	Retention and incentive
I	3.3.1	Organise learning process for the students
J	3.3.2	Activity planning for teachers
K	3.4	Usage and progress statististics
L	3.5.1	Collaborative production of educational resources
M	3.5.2	Teacher – student forum
N	3.5.3	Support among students
O	3.6.1	Training exercises

If you have a great idea that you consider useful and which is out of this list of functions, please refer to section 3.6.2.

3.1 Skill assessment

The following functions aim to enhance the assessment of students. In order to assess the mastery of a skill, Oscar prompts the student with questions.

3.1.1 Questions with adaptive answer boxes

Background :
The assessment of certain skills requires more sophisticated encoding than a free field or multiple choice list.

Summary :
A teacher may write a question with blank (or multiple choice) spaces that the student will be asked to fill in.

Examples :
1) Fill in a proportionality table

Distance covered (m)	70	105	280	
Elapsed time (s)	20	30		90

The student must be able to edit the boxes here indicated by the gray background.

2) Describe the construction of an angle bisector (here in French)

Je pose la ▮▮▮ sèche du ▮▮▮ au sommet de l'angle. Je trace ensuite un ▮▮▮ de cercle qui croise les deux ▮▮▮ de l'angle. [...]

Desired functions :
- In the teacher's interface: writing a question mixing formatted text, tables, images in which he/she puts some answer fields (open field or drop-down menu). The teacher encodes the correct answer(s) expected in each answer field. This encoding must be consistent with the one already in force for open-field questions.
- The student must be able to enter an answer, in the cell of a table or at a specified place in a sentence.

Extension of the basic function (to be started only after validation of the basic function by Eureduka): automatic generation of such questions. The module picks up a model question statement and produces random numbers satisfying certain constraints to fill in the table. One or two cell(s) are left empty but the module stores the correct answer(s). A model question statement might be of the form : « A biker travels around at constant speed and writes down the distance covered (in km) and the time elapsed (in minutes) from time to time throughout the trip. The table below displays most of the information. Fill in the blank cells. » Obviously, the data displayed in the table should be realistic given the question statement !

Applications (for both the basic function and the extension) :
- Calculate an element of a proportionality table
- Build a table from a graph for a function of the first degree $x \to mx + p$ ($m \neq 0$)
- Build a table from a formula for a function of the first degree $x \to mx + p$ ($m \neq 0$)

Remarks :
- The blank boxes to be filled in must all have - within a sentence or table - the same dimensions, to avoid giving a clue to the student (who would understand

that a narrower box contains a shorter word/number). By default, the box adapts to the length of the longest content.

3.1.2 Questions aiming to assess drawing skills

Background :
The assessment of certain skills requires more sophisticated encoding than a free field or multiple choice list. In the current version of Oscar, it is possible for a teacher to create a question that assesses the positioning of points in a plane reference frame. The function described here is intended to extend this basic function to other kinds of drawings.

Summary :
The teacher can create a question to test a student's ability to draw (on a screen) a segment, ray (or half-line), line, polygon, circle or combination of these elements.

Desired functions:
- While creating a new question: the teacher presses a button representing a reference frame in the plane.
- An empty reference frame appears. The teacher draws the figure and records it as a correct answer.
- Extension of the basic function (to be started only after validation by Eureduka): automatic generation of questions of the type:
  - « Set the coordinate point (x, y) in the reference frame below » where the teacher only determines the constraints on the allowed values of x and y (by default: x between -5 and 5 and y between -5 and 5).
  - « Draw the equation line $y=ax+b$ in the reference frame below ». In this example, the teacher only determines the constraints on the type of coefficients (entire, fractional or decimal) and their allowed values (by default: a between -3 and 3 and b between -5 and 5).

Remarks :
- To place points on a small screens, consider "point, drag and drop" to reduce the risk of ill-positioning.
- Think carefully about the correction following the ill-positioning of a point: remove then place again, or point-drag-lift-release?
- To draw a half-line, first point the end and then the waypoint.
- For the positioning of a straight line or a half-line, admit a difference in the waypoints (between the ones chosen by the teacher and the ones chosen by the student): what counts is the position of the (half-)line.

3.1.3 Questions aiming to assess statistical graphics skills

Background :
The assessment of certain skills requires more sophisticated encoding than a free field or multiple choice list. The function described here is intended to extend this basic function to other kinds of drawings.

Summary :
The teacher can create a question to test a student's ability to plot (on a screen) or adapt
a statistical graphic : bar chart, frequency polygon, pie chart, histogram, scatterplot.

Desired functions:
- While creating a new question: the teacher presses a button representing one of the statistical graphics.
- The appropriate graphic type appears. The teacher enters the data and records it as a correct answer.
- Extension of the basic function (to be started only after validation of the basic function by Eureduka): automatic generation of questions of the type:
  - « Draw the pie chart corresponding to the table below » where the teacher specifies that the angles at the centre of the sectors are whole values in degrees or multiples of 2°, 5° or 10° (by default: multiples of 5°). It also writes the allowed number of sectors (by default: random between 4 and 6).
- For the construction of a pie chart, the user sees a first sector appearing with an angle in the center of 20°, equipped with a handle to modify the amplitude of the circular sector.
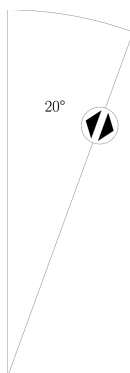
- When the first sector is set up, the user can add a second sector (button displaying a sector and a "+" symbol), whose angle at the centre is still 20°. Of course, it can change its amplitude. The cursor between the first two sectors remains visible. Generally, when the user moves a cursor between two sectors, both sectors are modified, keeping the outer edges remain unchanged (i.e., the sum of the amplitudes of the two sectors remains constant).
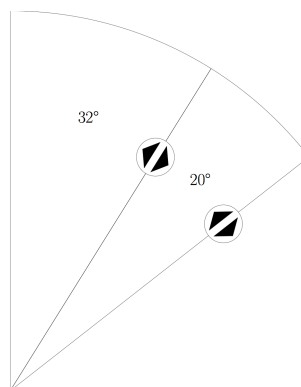
- If the available amplitude (empty space for extra sectors) is less than 20°, the student can not add a new sector: this would help the student who has not calculated the amplitude correctly. Oscar should ask him what amplitude he wants for the next sector. If the mentioned amplitude leads to more than 360° for the pie chart, a warning appears and the display returns to the previous chart (without added sector and with sectors amounting less than 360°).

3.1.4 Evaluation of algebraic expressions

Summary :

In order to test a student's ability to solve an (in)equation, it is desirable not to limit oneself to the solutions alone, but to observe the steps of reasoning. We ask you to create a mathematical expression evaluation engine and (by extension) a mathematical expression comparison engine.

Desired functions:
- Validation of the equivalence of two expressions on either side of a symbol "="
- Comparison between two steps of a resolution: is the second step deduced from the first?
- This function has two versions, depending on the use :
  - (Default for student assessments.) The student completes all the steps of his/her resolution of (in)equation on piled up lines. He/she validates at the end and then sees at which step(s) he/she has written incorrect expressions or which are not equivalent to the previous ones.
  - (Default for student training.) The student submits each line of his/her resolution and immediately sees if he/she has made an error, which can then be corrected. Only when one step is correct can the student complete the next step.
- There is a comment in the margin that points the kind of error at stake at that step.
- This function should cover six question statements :
  - Resolve an equation of the first degree with one unknown (appearing only on one side of the equation)

- ○ Resolve an equation of the first degree with one unknown (appearing on both sides of the equation)

- ○ Resolve an inequation of the first degree with one unknown (appearing only on one side of the inequation)

- ○ Resolve an inequation of the first degree with one unknown (appearing on both sides of the inequation)

- ○ Resolve a system of two equations of the first degree with two unknowns (each appearing only on one side of each equation)

- ○ Resolve a system of two equations of the first degree with two unknowns (each appearing on both sides of each equation)

- Extension of the basic function (to be started only after validation of the basic function by Eureduka): automatic generation of questions of the six types mentioned above. In all these cases, the teacher must be able to define the allowed values of coefficents and/or answers (range and kind : naturals, integers, fractions). The teacher can also change the name(s) of the variable(s), which is/are x (and y) by default. The teacher will finally decide to cover or not impossible or undetermined (systems of) equations.

Technical constraints :
This function requires symbolic manipulation. Feel free to use a symbolic manipulation library in Python.

3.1.5 Automated question/answer creation (one group or two groups)

Background :

So far, each assessment question has to be written by a teacher. There are - to date - few questions for a given skill. A student with an unmastered skill could be faced with the same question associated with that skill.

Summary :

We wish to generate a large number of similar questions on the basis of a framework.

Desired functions:

- Automatic creation of questions to assess the skill "Determine the root(s), if any, of the polynomial ax^2+bx+c". The teacher determines the allowed values of a, b, c (natural, integer or rational) and the allowed values of the answers (integer, rational, complex with real and imaginary parts both integers, complex with fractional parts). Note that the student can type answers with fractions through a keyboard displayed on screen (function already available).

- Same for other kinds of questions :

- ○ Calculate the perimeter of a triangle

Oscar

- Calculate the area of a right-angled triangle

- Calculate the area of a triangle (excluding right-angled triangles). In this kind of question, one altitude of the triangle should appear in vertical position.

- Calculate the area of a triangle (excluding right-angled triangles). In this kind of question, one altitude of the triangle should appear but NOT in vertical position.

- Calculate the perimeter of a circle

- Calculate the area of a circle

- Calculate the perimeter of a quadrilateral (seperate questions for rhombus, square, rectangle, parallelogram, trapezium)

- Calculate the area of a quadrilateral area (seperate questions for rhombus, square, rectangle, parallelogram, trapezium)

- Calculate the perimeter of a regular polygon (n=5 to 10)

- Calculate the area of a regular polygon (n=5 to 10)

- Calculate a length using Pythagoras theorem

- Calculate characteristic values of a set of statistical data (one characteristic value for one question)

- Calculate a simple interest

- Calculate a compound interest

- Combine different representations of a same object (different questions for cylinder, cone, prism, pyramid)

- etc. (consult the Oscar team for a complete list or for more information on these skill assessments)

- In all these cases, the teacher must be able to define the allowed values of parameters and/or answers.

Remarks :

This function might be assigned to two groups. Their members can freely collaborate. Working together or not, they have to distribute the requested types of questions among the groups and cover most of the desired tkinds of questions.

3.1.6 Scan and import of handwritten answers

Background :

In some classrooms, there is no computer equipment. As far as possible, we want to make the Oscar features and resources available to those classrooms. In addition, it is

difficult or impossible to automatically process the assessment of certain complex skills (such as writing a demonstration or justification) online. Instead of creating an online test, the teacher can choose to print a PDF file for a written evaluation in the classroom. This function is already available today, in a simplistic version.

Summary :

Handwritten replies should be retrieved as far as possible as scanned images in the Oscar site. Some answers might even be interpreted (as correct or incorrect) automatically.

Desired functions :

- Transmission of student copies via scanner to a processing module that extracts students' answers (written within boxes) and automatically associates them with the questions asked.

- Online display of scanned answers, teacher annotations and correct answer. Each student sees each of his/her answers, associated with the appropriate question, and the correct answer by clicking on the test link. The teacher can browse the individual copies, either through a student's profile and then ask for the test details, or by clicking on the test (at class level) and then asking for the details per student.

3.1.7 Questionnaire/survey towards students or teachers

Summary :

It seems that the learning experience of mathematics can be improved for many students through emotional awareness and metacognition.

On the other hand, it might be useful to survey teachers or students about their level of satisfaction and practices.

Desired functions :

- To help students take a step back from the learning or assessment experience, we would like to provide them with questionnaires in text or symbolic form (e.g., in the form of smileys representing their feeling/emotion at the beginning of a test, then at the end of the same test).

- The administrators can organise a survey towards students or teachers. They can specify which students/teachers it is addressed to : survey for a class or for all students in a grade, etc.

- A teacher can organise a survey for his students. They can specify which students it is addressed to : survey for a class or for all students in a grade, etc.

- The administrator or the teacher chooses at what time the survey appears : at login (only choice for teachers participating in a survey), at beginning of next test, at end of next test, at beginning of all upcoming tests, at end of all upcoming tests.

3.2   Retention and incentive

3.2.1 Rewarding teacher efforts (grouped with 3.2.2)

Summary :
We are confident that teachers are willing to help their students learn better. We would like to encourage them to share their work and collaborate. We want to set up a contribution rewarding system (depending on quantity and quality), similar to the « reputation points » and « gold medals » of *stackexchange.com* and so many other technical forums.

Desired functions :
- Teachers can obtain a status by contributing (=adding questions and ressources available for the whole community)
  - Contributor (addition of 1 resource that scored 4-5 stars)
  - Motivated contributor (addition of an extra 3 resources that scored 4-5 stars)
  - Advanced contributor (addition of an extra 5 resources that scored 4-5 stars)
  - Super contributor (addition of an extra 10 resources that scored 4-5 stars)
  - Top contributor (highest number of resources that scored 4-5 stars, with a minimum of 20 resources added with score 4-5)
- When a teacher gets a status, a pop-up appears "Congratulations, you have added X resources with 4-5 stars. You are now a <status name>! "
- Each status is linked to an icon that is next to a resource added by a teacher. When you hover your mouse over the icon, a scroll-over appears with the phrase: "Added by a <teacher's status>".
- The teacher keeps his/her status (except « top contributor ») even if some of his/her added resources get down below 4 starts after some time.

3.2.2 Evaluation of resource quality (grouped with 3.2.1)

Summary :
Every student, every teacher can express his or her appreciation of an educational resource. A vote is 0 to 5 stars, with the possibility to add comments. (This is very similar to what you find about apps in GooglePlay or AppStore.)

Desired functions:
- Vote by students and separate vote by teachers for each educational resource (like apps for smartphones). The counting of votes is calculated on the one hand for all pupils and on the other for all teachers. Everyone sees the average and the vote count. (This is the only function that is truly public.)
- A teacher is not allowed to vote for his/her resources ; his/her students neither.
- The administrators can manage comments (at least : display all – hide all) to avoid abuse.

3.3 Work organisation

### 3.3.1 Organise learning process for the students

Background :

At login, a student sees a couple of so-called « recommended » skills. These are the unmastered skills whose prerequisites are all mastered (or which do not have any prerequisite), meaning they constitute the next step in the learning process. This information do not favour a skill over the other ones : the student has to pick one among the recommended skills, sometimes based on the teacher advice.

Summary :

This module helps each student manage his work in order to reach the goals set by the teacher, by taking its cognitive situation into account. It points out which recommended skill he/she should start with.

Functions :

- Each skill gets an estimated time to reach mastery : 5 time slots (5 x 50') by default.

- The teacher can define « targets ». These are skills the teacher expects the student to master in the near future. In order not to put too much workload at once for the student, the teacher can define at most 3 « targets » at a time. The figure below shows the skills mastered by the student (green) and the skills unmastered by the student (orange). The targets are unmastered skills. In this example, both skills at level 5 are defined as targets. (Targets don't have to be at the same level though.)
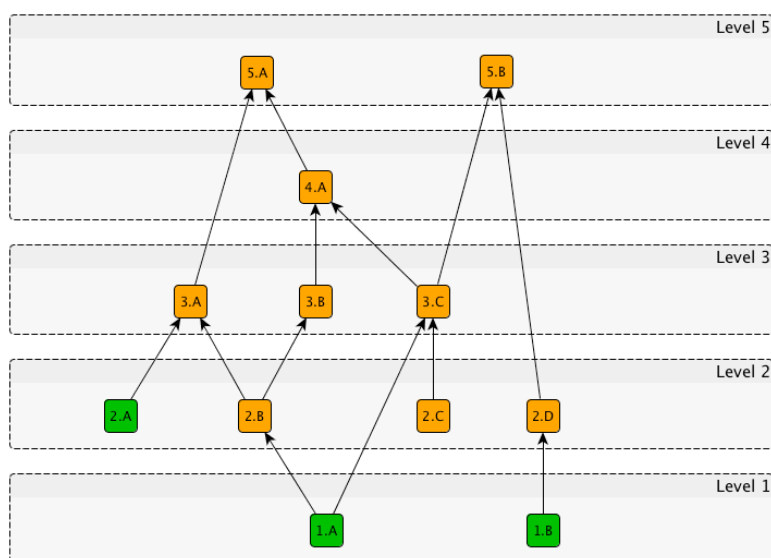
A sample graph of skills. Each node represents a skill and

each arrow means « is a prerequisite of ».
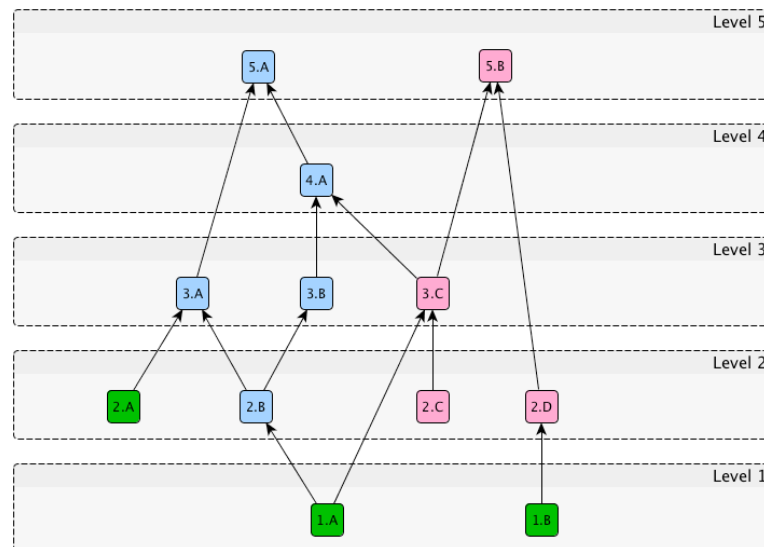
The same graph showing the groups of skills (blue and pink resp.)

for the unmastered and untested skills.

- Oscar enforces an order on the skills (the recommended skills and the upcoming skills) and points out the one the student should start with. The ordering criterias are the following by default :

  ○ position in curriculum : the « oldest » or most basic skills are handled first. In the sample graph above, skills at level 2 should come before skills at level 3 ;

  ○ skill group : the skills from the same group should be handled just after one another as much as possible. In the sample graph above, we will avoid the sequence 2.C-2.B-2.D or 2.D-2.B-2.C that would split the skills 2.C and 2.D from the same group.

  ○ (if not in contradiction with previous criterias) estimated to time to reach mastery : the skills that require less time to reach mastery should come first, in order to let the student get « quick wins ». At this stage, we still have the choice between 2.B-2.C-2.D or 2.C-2.D-2.B. If 2.B and 2.C do not have the same estimated time to reach mastery, we start with the shorter among the two. (If they have the same estimated time to reach mastery : pick up one at random.)

- Oscar displays the recommanded « first skill to master » based on the order enforced on the skills. Oscar shows also what's coming next. All previously mastered skills should also appear (potentially as grayed out) as « past victrories ». Showing only the skills that are not mastered might be demotivating for the students.

- The teacher can modify the proposed « learning track », before being passed on to the student. The order enforced by the teacher would then « overwrite » the order made up from the criterias listed above.

Technical constraints :

It is expected that the algorithm enforcing the ordering criterias is easy to change by administrators. Even better : these ordering criterias could be changed through an user interface.

3.3.2 Activity planning for teachers

Background :

Teachers must organise their work on several time scales : respect for the required curriculum before the end of the school year, and the organisation of activities, from week to week (lectures, activities and assessment mostly).

Summary :

We want to develop an interactive interface in which the teacher drag-and-drop types of activities into a (nearly) blank school calendar.

Desired functions:

- The teacher proceeds in two steps. At the beginning of the year, he completes his weekly schedule. During the course of the year, he organizes, for each class, the sequence of activities (courses, activities, outings, evaluation, etc.).

- The teacher, if he/she wishes to use this function, will complete his/her weekly timetable, specifying the hours[1] and classes, as follows:

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
8h	6TQA	6Transition	-	6Transition	-
8h50	6TQA	6Transition	-	6Transition	-
9h40	-	-	-	-	-
10h45	5TQA	6TQE2	-	6TQE2	-
11h30	5TQA	5TQE1	-	5TQE1	-
12h20	-	-	-	-	-
13h10	-	5PA	-	5PA	-
14h	-	6PA	-	6PA	-
14h50	-	-	-	-	-
15h40	-	-	-	-	-

- A drop-down menu in each box, indicating the classes previously created by the teacher. (If not, ask the teacher to create his or her classes first.)

---

1   By default, the mentioned times will be valid, but the teacher can modify them via a configuration interface.

- Possibility to change the start times of the course. By clicking in a cell containing a time (left column above), the information becomes editable. The teacher validates or click anywhere out of the editable box. The new time is accepted and saved as long as the information entered has a correct format and is higher than the previous one.

- The annual calendar will be prepopulated to facilitate the teachers' work as much as possible: public holidays and school holidays will appear. For the years 2017-2018 and 2018-2019, the holidays are available here:
  http://www.enseignement.be/index.php?page=23953

- Teachers modify the calendar obtained: adding pedagogical days (with no lecture), school trips, etc. They then see the calendar of days actually reserved for teaching activities.

- There are 3 distinct periods over a school year : - September to December - January to March - mid-April to June. The teacher can display the full school year, a period, a month or a week.

- By class: you can display a summary view covering a period, month or week.

- By class: blank canvas in which the teacher indicates the activities he/she intends to do. Drag and drop activity types into an (almost) blank calendar. (Holidays, pedagogical days, school trips, etc. appear as unavailable: no activity can be organised there.)

- The dates for classroom assessments are encoded in such a way that the students concerned are notified (see the work by another group, section 3.5.2).

3.4 Usage and progress statististics

3.4.1 Tracking student activity and progress (grouped with 3.4.2)

Summary :
A teacher gets a graphical overview of the status and ongoing progress at the level of a class or at the an individual level.

Desired functions :
- One tab with an overview of all students, one tab for each class. In each tab, there is a dashboard.

- Dashboard showing usage statistics (low importance), skill mastery level and progress made as a whole and (only in the « class » tabs) for each student. Examples : number of skills mastered per week, time spent since last mastery of a new skill, etc.

- The teacher can change the timespan for the timeline statistics.

- Within a « class » tab, one can get access to even more detailed information about a student by clicking on his/her name.

- Possibility to export the raw data obtained after applying the filter (whose summary is displayed - see previous point), within the specified format (at least the "normal" CSV format with "," as separator mark and "." as decimal mark and the "European" CSV format with ";" as separator mark and "," as decimal mark).

3.4.2 Data useful for research (grouped with 3.4.1)

Background :

Laurent Fourny is conducting research at UCL to improve the identification of a student's state of competence. This research is based on statistical data. The behaviours and profiles of users could be very instructive in this regard.

Summary :

The site administrators have access to an interface allowing them to graphically observe certain information and download the raw data.

Desired functions :

- Dashboard showing some global information (detailed by level of education, gender): number of users, number of new users over the last month, last week, average connection time, etc. Information on adding questions and resources.

- Same information about a sample obtained after applying a filter (time period, grade, etc.).

- Possibility to export the raw data obtained after applying the filter (whose summary is displayed - see previous point), within the specified format (at least the "normal" CSV format with "," as separator mark and "." as decimal mark and the "European" CSV format with ";" as separator mark and "," as decimal mark).

3.5 Collaboration

3.5.1 Collaborative production of educational resources

Background :
Most teachers work alone. They design course materials, which only reach a satisfactory quality after years of work. Collaboration between a few teachers - like the teams that write textbooks - would be doubly beneficial. On the one hand, they would produce the same type of support (much) faster than if they had worked alone. On the other hand, skills and knowledge can complement each other, and ideas mature further when interacting with colleagues.

Summary :
This function aims to encourage exchanges and meetings between teachers who are supposed to develop the same kind of educational resources (at a minimum, they have

classes at the same level of education). It should also enable remote collaborative work with a minimal document management platform (uploading/downloading/commenting). Once the work is completed, team members can customize the finished product to suit their respective classes. We encourage them to place the anonymous version of the finished product on the Oscar resource platform ('wiki').

<u>Attention</u> : this function is not limited to mathematics teachers!

Desired functions :
When using this collaborative service for the first time, check the teacher's available data. If they are incomplete, invite him/her to complete them. In addition, the teacher is asked if he/she wishes to develop an educational resource with colleagues ("proposal submission"). In any case, (s)he can observe the proposals for collaboration made by colleagues in their notifications and in the « collaborate » interface (to be designed, with a map).

Submission of a proposal: the teacher fills in a form specifying what he/she wishes to develop (with, as far as possible, the opportunity to say "at will") : area (single discipline or cross-disciplines), learning object (one skill or a group of skills), grade, type/format (video, activity, PDF, learning sequence, etc.), wishes/comments (open field), availability for meetings (weekdays), etc.
The proposal may be a targeted resource, a group of skills (= a chapter) without a preconceived idea of the resource, or a broader project (including interdisciplinary activities). In the latter case, the initiating teacher can create a page with the proposed division of work, which colleagues can choose to complete.
A collaboration proposal appears in a teacher's notifications/messages if certain conditions are met. Default: within scope of teacher expertise, less than 30 km between collaborators, same teaching year ±1. If this is a targeted resource, proposals will only appear if the teacher is working in the subject concerned. In the case of an interdisciplinary project, teachers in the disciplines covered receive the call.

Reaction to a proposal: this part is left to your imagination.

3.5.2 Teacher – student forum

Summary :

We would like to add a messaging module, to enable a teacher and his/her students to interact online.

Desired functions :

- The student can contact his teacher at any time to ask questions.
- The student may write a remark about a resource that is inappropriate or difficult to understand, a question that is confused or inappropriate for the skill, a mathematical notion that is difficult to understand (even if the educational resources seem appropriate and clear). Depending on whether or not the remark may be of interest to other students in his/her class, he/she sends a private

message or a "public" message restricted to his/her class. By default, the message is only sent to its teacher. The teacher who receives a comment from a student can pass it on to the Oscar team with the possibility of writing a comment. The teacher can also send such a message spontaneously. This is especially true for resources/issues that a student or teacher may find inappropriate.
- Each message can be associated with a skill or group of skills (or a UAA[2] for students in grades 3-6 at secondary school).
- When the teacher receives such a message, an alert appears in his/her interface.
- The student can also take a picture of a drawing that he/she would have made and add it to a message before submitting it to the teacher.

Technical constraints :
- The Oscar team's email cannot appear in plain text in the code, to avoid being spotted by a robot. Use the various procedures used in online contact forms to hide the contact email.
- A messaging interface is created with fields:
  - a drop-down menu: object of the message (general question, remark or question about a resource, remark or question about a skill, remark or question about a skill group - or an UAA for students in secondary school, at grades 3-6) ;
  - depending on the object chosen, a list with a search bar appears to search for either a skill, resource or skill group... etc.) ;
  - a text field to write the message ;
- An icon (to be defined) is visible in the student interface next to each skill, or skill group. When the student clicks on it, the message interface appears with all the fields filled in with the information depending on where he clicked (except for the text field, left empty).

3.5.3 Support among students

Summary :
- This module should allow students of the same level and region to meet to help each other. Reserved for students from grade 3 at secondary school onwards.
- Draw inspiration from the collaborative module for teachers (section 3.5.1).

3.6   Miscellaneous

3.6.1 Training exercises

Summary :

Most of the Oscar website is currently about skill assessment. It lacks training exercises. We would like to develop that part of the website and ensure it is as fruitful and enjoyable as possible for the students. In this case, we assume they don't get any support from their teacher.

---

2   « Unité d'acquis d'apprentissage » can be considered as a consistent set of skills. They would be grouped within the same chapter in a textbook.

See the « matching pairs », « number line » and « multiple-choice quiz » apps made available on https://learningapps.org for inspiration, and provide the teacher with an user-friendly interface to create such exercises.

3.6.2 Module at will

Attention : unless you have a great idea in mind, this function is only available if the development of your main module is fully available and integrated. Anyway, you should get explicit agreement by Prof. Kim Mens and the Eureduka team representative to start working on this module.

Summary :

You have an idea, building upon the current website, to enhance the experience of students and/or teachers. You describe it, you present it for validation, and you build a prototype. We don't expect a full integration in this case.