Requirements

Requirements are based on a business model of software as a service. An instance of the system is one unique school.

Teacher

UC1: Manage Classroom

UC2: Manage Syllabus

UC3: Manage Exercises

UC4: Follow Up on Students

UC5: Manage Assignments

UC6: Manage Activities

UC7: Manage Lessons

UC8: Post Class Announcements

UC9: Chat With Students

UC10: Receive Statistical Information (Individual student performance, exercise quality)

UC11: Manage Theoretical Content

Student

UC12: Join Class

UC13: Interact with Students (chat)

UC14: Manage Self Profile (includes exercise, assignment and activity history and general stats)

UC15: Solve Exercises

UC16: Submit Assignments

UC17: Play Group Activities

UC18: Request Help with an Exercise – live and automatic

UC17: Interact with Teacher

Design and Architecture

**Store every action a student makes.**

Support inactive content – content that was prepared by users, stored but not added to a class. (self notes).

Support internationalism of user interface, with no respect to the language of the class content.

Separate between persistence of static and dynamic content.

Data must be always available.

Support free text search of exercise.

Prepare for future implementation of a translator and media dictionary.

Consider future options to practice Spoken language.

Interaction between players in a single game should be fast and accurate.

Strong authentication and authorization mechanisms in separate components.

Define assignments as an action one student makes and some variable setting the interacted object exposes as dynamic.

Consider high dynamic content capabilities.

**UC1**

**Actor:** Teacher

**Precondition:** User is logged as Teacher

**Manage Classroom:**

1. New Classroom: Teacher selects a Language of interest. Assigns a name to the class and a short description. The details of the teacher are assigned to the new class. Creates new lessons as defined in UC7.
2. Existing Classroom: (cross ref UC2-7)

**UC2**

**Actor:** Teacher

**Entered From:** Manage Classroom

**Precondition:** User is logged as Teacher, classroom exists and is selected.

**Manage Syllabus:**

1. Create New:

Precondition: User does not already have an existing syllabus (only one permitted per class)

User assigns a description to the syllabus and an optional name. Repeats the following for each Lesson:

* 1. Assigns a name and a short description
  2. Adds theoretical content
  3. Adds exercises (cross ref UC3)

User Sequences all lessons by a chosen timeline. Defines minimal and bonus content per lesson as different exercises that must be completed. Defines the last lesson as the end of the syllabus (optional).

1. Manage Existing: User can select to change the name of the syllabus and description. Can do one of the following:
   1. Add lessons
   2. Disable Lessons – make a lesson obsolete, but still accessible to students
   3. Define new ending for the syllabus

**UC3**

**Actor:** Teacher

**Precondition:** classroom exists.

**Manage Exercises:** User selects exercise by specific criterion, or uses search to find relevant ones.

1. Create New: User selects one or more lessons as the subject of the exercise. Enters an optional name and short description. Selects one of the available formats.

Adds Paragraphs to exercise – process differs by exercise format (UC3.\*s).

Defines the minimal criteria (minimal level of execution) to pass an exercise as the percentage of correct answers from the entire exercise.

Optional – defines additional criteria as before and assigns them different levels of execution by his interpretation.

1. Modify Existing: User selects an exercise to modify, manages questions within exercise. Changes

UC3.1

Create Fill-In the Blanks Exercise:

1. User selects answering scheme – “multiple answer options” or “self-input” and repeats the following:
   1. Creates a new paragraph:
      1. Enters text content disregarding any missing content
      2. Marks points in content where missing fields should appear.
      3. For each such field, adds all possible correct content
      4. If answering scheme is multiple answer options
         1. Selects number of possible options
         2. Adds “dummy” content to each blank field until possible options count is as specified.

Else if answering scheme is “self-input”, do nothing

* + 1. Defines the weight of a question in the context of the exercise (alternatively may be done after all paragraphs are added).
    2. Optional – may supply a clue for each question as a text sentence.
    3. Optional – may supply a short explanation for each paragraph.
  1. Sequences the paragraphs by order of appearance.
  2. For each paragraph, may mark it as Bonus. Alternatively, may create a bonus exercise where all paragraphs are such.

UC3.2

Create Simple Question-Answer Exercise:

1. User selects answering scheme – “multiple answer options” or “self-input” and repeats the following:
   1. Creates a new paragraph:
      1. Enters questions content
      2. Supplies all possible correct answers for the question.
      3. If answering scheme is multiple answer options
         1. Selects number of possible options
         2. Supplies “dummy” answer content.

Else if answering scheme is “self-input”, do nothing

* + 1. Defines the weight of a question in the context of the exercise (alternatively may be done after all paragraphs are added).
    2. Optional – may supply a clue for each question as a text sentence.
    3. Optional – may supply a short explanation for each paragraph.
  1. Sequences the paragraphs by order of appearance.
  2. For each paragraph, may mark it as Bonus. Alternatively, may create a bonus exercise where all paragraphs are such.

UC4

Actor: Teacher

Precondition: Teacher selected a student

Follow-up on Students: Teacher sees history of all questions answered by the student, including time and execution level. The history and current state of all assignments is presented to him.

UC5

Manage Assignments:

1. Create New Assignment: Teacher enters name, short description, and relates the assignment to a lesson. Repeats the following:
   1. Select a student action (currently only solve exercise is considered)
   2. If Solve Exercise is selected
      1. Select exercise from library
      2. Selects level of execution the exercise must be solved at (e.g. A+,A,C etc.) from ones defined to exercise.

UC7

Actor: Teacher

Precondition: User has selected a Class

Manage Lessons

1. Create New Lesson: teacher enters name of lesson and a short description. Creates appropriate theoretical content and adds any number of exercises for the lesson.

Create “Remove The Unnecessary” Exercise without multiple option answer:

1. User selects format and repeats the following:
   1. Creates a new paragraph:
      1. Enters content in its minimal correct form
      2. Marks points in content where additional content should appear.
      3. Defines up to 4 multiple answers as a set of missing contents matching blank fields, by order.
      4. For each answer set, defines if it is the correct one or not. Number of correct answers must be low than the number of possible answers.
      5. Defines the weight of a question in the context of the exercise (alternatively may be done after all paragraphs are added).