Contents

Chapter 1

Use cases

1.1 Actors

The platform recognizes two type of actors in the first place, the unidentified user and the registered user.

1.1.1 Unidentified user

This actor has only registration and authentication rights on the platform.

1.1.2 Registered user

A register user has only one identity listed below.

Student

This actor has the rights to take exercises and be graded by the platform.

Teacher

This actor has the rights to supervise students such as providing exercises.

Administrator

This actor has the rights above and make sure things are running smoothly.

1.2 Unidentified user

1.2.1 Overview

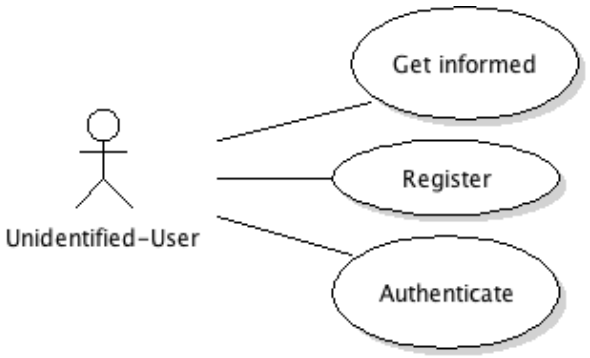

Figure 1.1: Unidentified user use cases : Overview

Get informed

An unidentified user gets informed about what the platform is.

Register

An unidentified user who wants to access to the platform has to register first.

Authenticate

An unidentified user authenticates to have access to the platform if the user is already registered.

1.3 Student

1.3.1 Overview

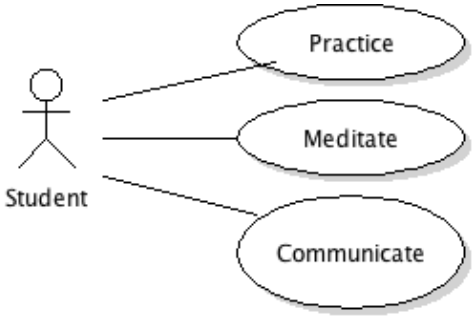

Figure 1.2: Student use cases : Overview

Learn

The student learns her lessons and do exercises.

Meditate

The students meditates on her performances and overcomes her weaknesses by doing suggested exercises.

Communicate

The students communicates with authenticate users in order to fix her issue and to improve herself.

1.3.2 Learn

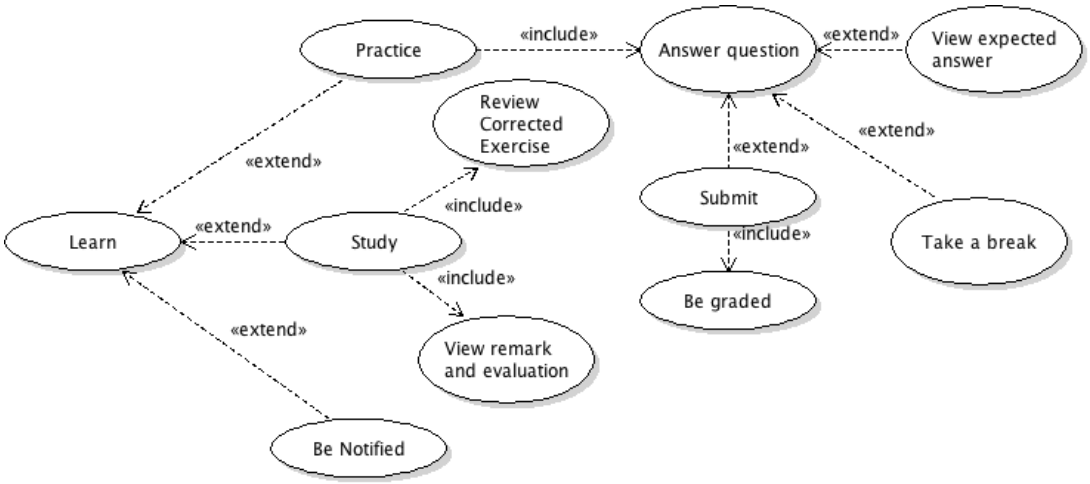

Figure 1.3: Student use cases : Learn

Be notified

The student is notified for new events concerning her learning process.

Study

The student studies by reviewing corrected exercises.
The student studies by reviewing teacher's evaluation and remarks.

Practice

The student practice her skills by answering questions.

Answer question

The student answers a set of questions.

View expected answer

The student has optionally access to the corrected version.

Take a break

The student takes a break and will resume the exercise later.

Submit

The student submit her answer form.

Be graded

The student is also graded by the platform *after* she submits her answer form.

1.3.3 Meditate

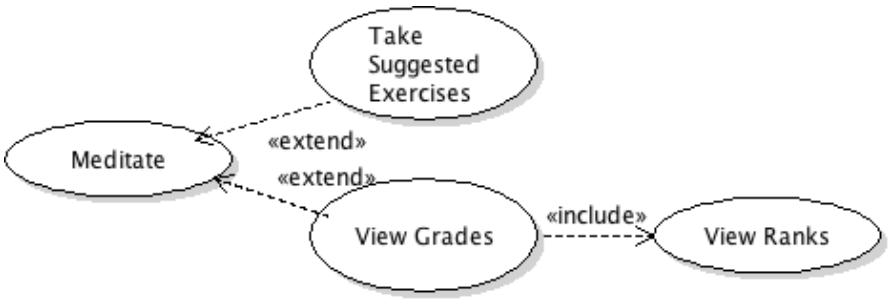

Figure 1.4: Student use cases : Meditate

Take suggested exercises

The student will be proposed exercises according to her grade.

View Grades

The student observe her grade for each exercises.

View Ranks

The student compares herself with other students.

1.4 Teacher

1.4.1 Overview

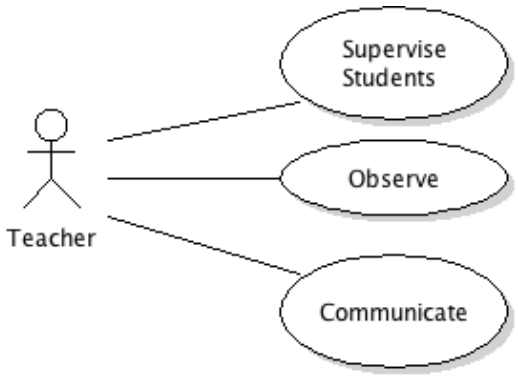

Figure 1.5: Teacher use cases : Overview

Supervise students

The teacher supervises students by providing exercises and grading them.

Observe

The teacher observes students's grade and remediate to help them.

Communicate

The students communicates with authenticate users in order to fix her issue and to help students.

1.4.2 Supervise students

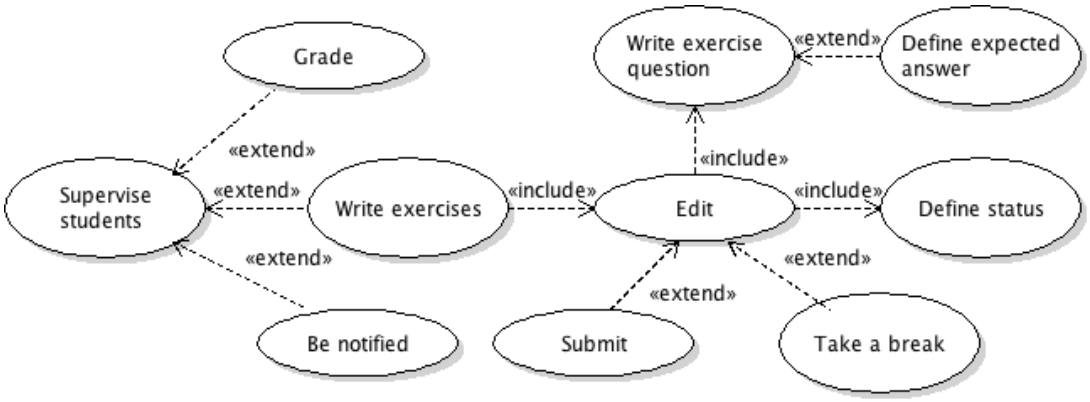

Figure 1.6: Teacher use cases : Supervise students

Grade

The teacher grades students answers for questions which have no fully-automatic grading system.

Be Notified

The teacher will be notified of events. For instance the teacher will be notified when one of her exercises is online.

Write exercises

The teacher writes exercise descriptions (statements, resources, ...).

Define expected answer

When the teacher writes exercises, he can optionally define an expected answer.

Edit

The teacher can edit any exercise at anytime.

Define Status

The teacher attaches a status to an exercise (date of release, condition of release, student concerned...)

Take a break

The teacher can resume the edition of her exercise later.

Submit

The teacher submits her exercise which can be viewed by student according to the exercise status defined by the teacher.

1.4.3 Observe

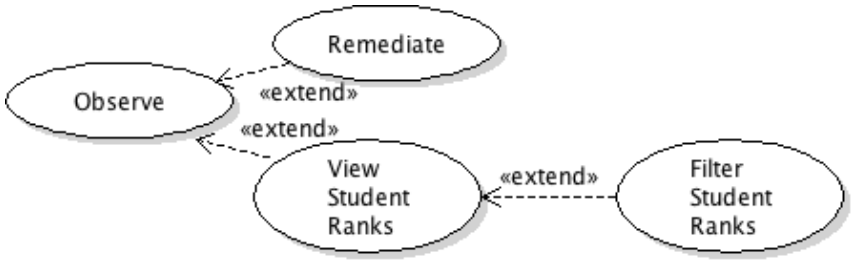

Figure 1.7: Teacher use cases : Observe

View students ranks

The teacher observes all the grades of students who took his exercises.

Filter students ranks

The teacher filters the grades according to some critera. The criteria can be the identifier of a group or the year for instance.

Remediate

The teacher can provide exercises for a group of students in trouble with some specific subject of the course according to their grades.

1.5 Administrator

1.5.1 Overview

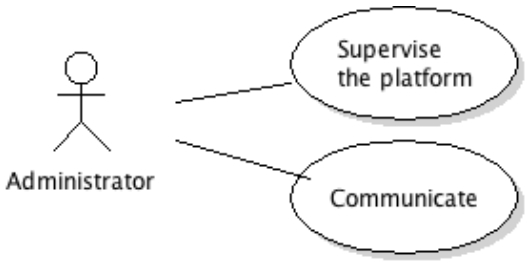

Figure 1.8: Administrator use cases : Overview

Supervise the platform

The administrator supervises the platform in order to ensure the platform operates smoothly.

Communicate

The administrator communicates with authenticate users in order to fix their issue.

1.5.2 Supervise the platform

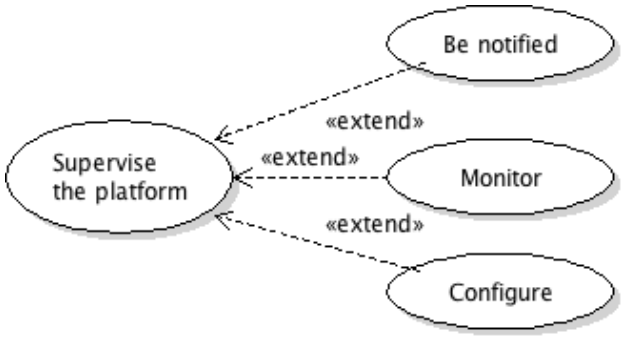

Figure 1.9: Administrator use cases : Supervise the platform

Be notified

The administrator will be notified by events such as a critical issue.

Monitor

The administrator inspects the log in real time.

Configure

The administrator configures the platform.

1.6 Common registered user

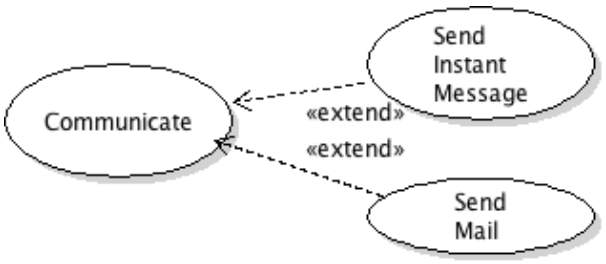

Figure 1.10: Registered user use cases : Communicate

Communicate

Identified user communicate between them through various message system provide by the platform.

Send Mail

Identifed user communicate by mail exchange.

Send Instant Message

Identifed user communicate by instant message exchange.

Chapter 2

Packages

2.1 Overview

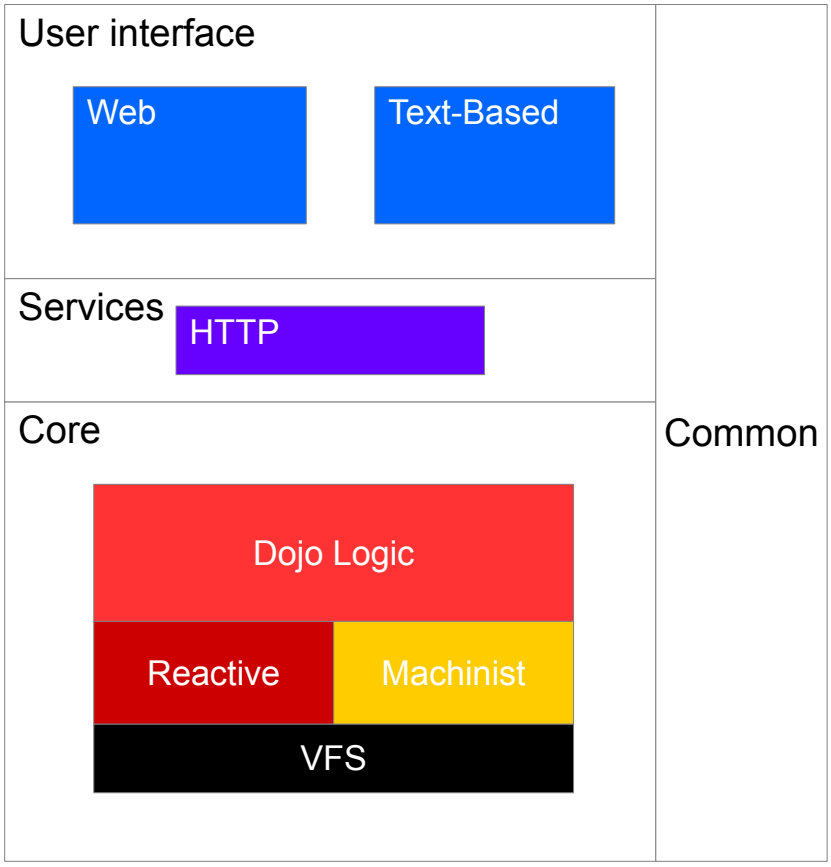

Figure 2.1: Diagram package : Overview

2.1.1 User Interface

This package provides every user interface components.

Web

This package contains the web user interface.

Text-based

This package contains the text-based user interface.
This interface interacts with unix-like batch tools.

2.1.2 HTTP

This package provides the web services which allow user interfaces to communicate with the core.

2.1.3 Core

This package provides the core features of the system i-e. the dojo logic and the machinist to make it works.

Dojo Logic

This package provides the business logic (online teaching system).

Machinist

This package provides modules which manage sandboxed environment.

Reactive

This package provides modules for data update through dependencies.

VFS

This package provides modules for managing the sources.

2.1.4 Common

This package provides utilities useful for every layers of the system.

Chapter 3

Modules

3.1 UI : Web

3.1.1 Exercise tools

This module contains every tools to edit an exercise.

3.1.2 Widget

This module contains various tool for a user friendly navigation.

3.1.3 Notifications

This module informs about every new information (new exercises, remaining exercises ...).

3.1.4 User

This module contains web view for user information (rank, profiles, mailbox ...).

3.2 UI : Text-based

3.2.1 Exercise tools

This module contains every tools to edit an exercise.

3.2.2 User

This module contains text-based view for user information (rank, profiles, mailbox ...).

3.2.3 Notifications

This module informs about every new information (new exercises, remaining exercises ...).

3.3 Services : HTTP

3.3.1 Redirection

This module links the UI layer with the Core layer though the HTTP services.

3.4 Core : Dojo logic

3.4.1 Exercise

This module orchestrates various other module to establish an exercise.

3.4.2 Description

This module is the language used to write an exercise.

3.4.3 Grade

This module contains the evaluation system and exercise correction.

3.4.4 User

This module contains user data (access rights, profiles, mailbox ...).

3.4.5 Data Analysis

This module analyzes

3.4.6 Notifications

This module informs users about their new or remaining task.

3.5 Core : Machinist

3.5.1 MachineProvider

This module is a sandbox provider.

3.5.2 Sandbox

This module is a closed environment given to the dojo logic, it can be from a virtual or a physical machine.

3.6 Core : Reactive

3.6.1 Entity

This module manages update and change between various exercise state.

3.6.2 Event

This module gives the type of event for an entity to be updated.

3.6.3 Dependency

This module handles dependencies between entity for chain reaction.

3.7 Core : VFS

3.7.1 Source

This module takes care of source file generated by the dojo logic layer.

3.7.2 Backup

This module makes a backup of every entity states.

List of Figures