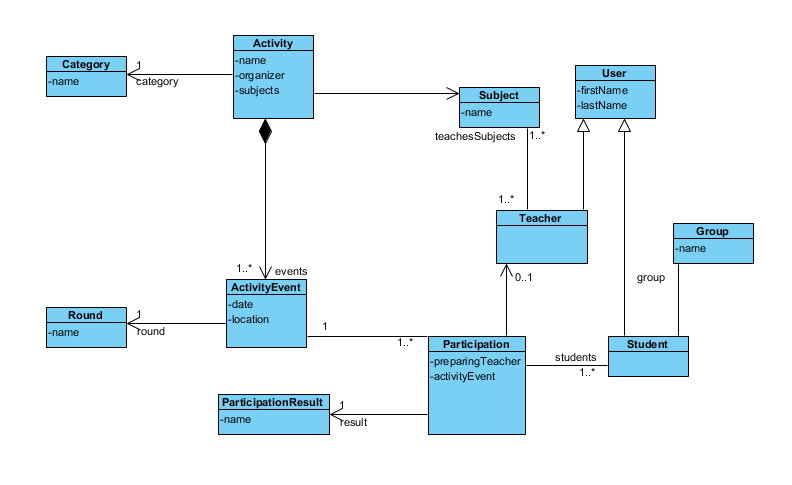
Student performance tracker

Analysis and Design Document

# Elaboration – Iteration 1.1

# Domain Model

The system records student participations in contests and activities. Each contest has a name, an organizer and belongs to a given category and can be about some subjects. A contest has one or more rounds, each round has location, a date when it took place. Students participate (individually or in teams) in one or more rounds of a contest, where they obtain results, students may be prepared for the round by a teacher.

Participations earn students and teachers points, depending on the category, the round and the placement (result).

# Architectural Design

## Conceptual Architecture

## Package Design

## Component and Deployment Diagrams

# 

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

## Class Design

Patterns:

* Value objects for Category, Round, Subject, Organizer, Group, ParticipationResult
* Builder pattern for the reports - reports can differ by the time interval they cover and the fields (columns) they include, and can take many parameters to build, which is a perfect fit for the builder pattern with a director.

# Data Model

# Test Strategy

## Unit tests

Testing for the service layer, with one test per method, like:

* Adding a new entity (Student, Teacher, Activity, Participation) is successful
* Awarded points are correctly calculated in the reports
* Data for existing entities is returned correctly
* Handling common invalid input

## Integration tests

Tests for the main use cases, like:

* The personal report of a student is correctly generated
* A teacher can confirm a student activity
* The application responsible can add/remove users.

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*