

Requirements Analysis Document

Lucía Asencio y Juan Riera

February 12, 2017

1. Introduction

- 1.1. Purpose of the system
- 1.2. Scope of the system
- 1.3. Objectives and success criteria of the project
- 1.4. Definitions, Acronyms and abbreviations

2. System Description

2.1. Functional Requirements The application has two types of users:

2.1.1. User type 1: student

2.1.1.1. Log in to the application

2.1.1.2. View available courses and send an application to a course
And get e-mail notifications when they are accepted or declined.

2.1.1.3. Access to each subject unit/subunit/subsub... and unit notes
Each item will only be visible after certain date (decided by teacher) if the student belongs to the course and is not expelled.

2.1.1.4. Access and solve unit exercises

Which will only be visible and solvable after and before certain date (decided by teacher)

Also, student may not answer a question if he decides so, and may quit the exercise without sending it whenever he wants, no changes saved.

Each exercise will be solved as much as once.

View his mark in each exercise, as well as correct answers (only after the deadline for the exercise has passed).

2.1.1.5. See their mark in a course

2.1.1.6. Communicate with teachers and other students
Via e-mail

2.1.2. User type 2: teacher

2.1.2.1. Log in to the application

2.1.2.2. Accept/decline students applications

2.1.2.3. Set up courses, unit, subunits, subsubunits...
Also, write and add notes and visibility to each unit

2.1.2.4. Use an exercise maker to add exercises to units
This maker will set the unit where the exercise belongs, the two dates between which it is visible, the number and type of questions (multiple choice/one choice/true-false/open answer), the order of the questions (random or not), the points of each questions, the penalty for each wrong answer, the value of the exercise for the final course mark.
Teacher can re-set up the whole exercise (excepting for the dates: only deadline can be changed and only if it is postponed), if no student has already done the exercise.

2.1.2.5. View statistics
Statistics per subject : marks, average mark, failed vs. passed for each subject.
Statistics per exercise : marks, average mark for each exercise.
Statistics per question : not answered, correct answers, wrong answers.

2.1.2.6. Expell students (and readmit them in the course)

2.1.2.7. Communicate with students
Via e-mail

2.2. Non-functional Requirements

3. Use Cases

3.1. Use Case diagram

3.2. Use case descriptions

3.2.1. Use case: Teacher
· Primary Actor
· Stakeholders and Goals

- Preconditions
- Success Guarantee
- Main Success Scenario
- Extensions
- Special Requirements
- Technology and Data Variations List
- Frequency
- Open Issues

3.2.2. Use case: Student

- Primary Actor
- Stakeholders and Goals
- Preconditions
- Success Guarantee
- Main Success Scenario
- Extensions
- Special Requirements
- Technology and Data Variations List
- Frequency
- Open Issues

4. Mockups