СУ “Св. Климент Охридски”,
Факултет по математика и информатика,
Катедра Информационни Технологии

Разпределена система за управление на курсове по програмиране с автоматично оценяване на решения

Здравко Иванов Гюров, РСМТ, ФН: 26379
Научен ръководител: Стоян Велев

София, 2022

Съдържание

I. Списък на съкращенията

API - Application Programming Interface

II. Речник на термините

Application Programming Interface - проложно-програмен интерфейс

III. Списък на фигурите

Asd

IV. Списък на таблиците

Asd

1. Увод

Обратната връзка за функционалната коректност, ефективност и придържане към конвенции за стил и добри практики е ключова за изграждането на знания и умения в курс по програмиране. Регулярната обратна връзка и особено възможността студентите сами да проверяват и подобряват итеративно решенията си, са възможни единствено чрез автоматизиране на процеса по тестване и оценяване. С увеличаването на броя практически курсове по програмиране и броя студенти през изминалите години, тези практики стават все по-наложителни за подпомагане на преподавателския състав да бъде

по-полезен и да предоставя персонална помощ за разрешаване на по-сложни казуси със спечеленото време.

Целта на дипломната работа е да се проектира и реализира разпределена система за управление на курсове по програмиране с автоматично оценяване на решения.

Структура на дипломната работа

В глава **2. Обзор на подобни системи** ще разгледаме няколко системи, които покриват изискванията ни до някаква степен. След това ще ги сравним и ще съпостави техните плюсове и минуси и най-накрая ще обобщим защо те няма да ни свършват работа.

В глава **3. Проектиране на системата** ще разгледаме функционалните и нефункционалните изискванията на системата и как ще бъдат изпълнени и предоставени.

В глава **4. Реализиране на системата** ще разгледаме конкретните технологии, с които ще бъде имплементирана системата и защо точно те са избрани. Ще видим също и основните API-та и как може един потребител да работи със системата

В глава **5. Експерименти и анализ на резултатите** ще разгледаме различни тестове, които са били приложени на системата за да се симулира реална работна среда и ще направим разбор на резултатите.

В последната глава **6. Заключение** ще направим обобщение, ще видим как системата ще влезе в употреба и ще разгледаме възможни подобрения и бъдещо развитие на системата.

Дефиниции

X наричаме У

2. Обзор на подобни системи

В тази глава ще разгледаме едни от най-популярните приложения и уеб услуги, които покриват възможно най-много от изискванията ни. Първо, нека определим кои са тези най-важни за нас функционални и нефункционални изисквания, които ще действат като критерии за сравнение.

1. Системата трябва да ни предоставя възможността да управляваме много на брой курсове (стотици, хиляди, а може и повече, за момента не се интересуваме от конкретен брой).
2. Всеки курс трябва да може да съдържа голямо количество задания (десетки, това отново е приблизително число).
3. Системата трябва да предоставя ясен и лесен начин за работа с предадените решения.
4. Системата трябва да дава възможност на преподавателския състав да проверява и оценява решенията по интуитивен начин.
5. Системата трябва да може да проверява решенията на студентите автоматично.
6. Системата трябва да може да улавя признаци на плагиатство у решенията на студентите.
7. Системата трябва да е гъвкава в отношение на технологиите, които се преподават в курса. Целта е да намерим една система, която може да се използва за всички ситуации, която ще е удобна за всички преподаватели и ще стане позната на студентите.
8. Системата трябва да дава свобода на лекторите сами да управляват курсовете си без намеса на администратор.
9. Системата трябва да е възможно най-достъпна финансово както за преподавателите, така и за студентите.
10. Системата трябва да е достъпна 24/7.
11. Системата трябва да е устойчива.
12. Системата трябва да е лесна за използване.

След като сме въвели тази основа, може да преминем към приложенията.
За този анализ са подбрани както системи, които са били в експлоатация дълго време и са се доказали като едни от най-добрите за времето си, така и новонавлезли системи, използващи модерни и иновативни подходи и технологии, а именно:

- Web-CAT
- codePost
- Microsoft Teams for Education
- Google Classroom
- GitHub Classroom

Да опиша всяка система, като покривам всяка една от 12-те точки отгоре и дали системата предоставя тази функционалност или не и накрая след 5те описания да направя табличка с 12те точки ясно да се вижда +/-

Подобни системи
Сравнителен анализ (+/-)
Обобщение - защо тези няма да ни свършат работа

3. Проектиране на системата

Да има диаграми, архитектурна, дб ,флоу

4. Реализиране на системата

Технологии, защо съм ги избрал
Да опиша основните апита
Снимки на ui-а

5. Експерименти и анализ на резултатите

Asd

6. Заключение

Постигнати резултати

Проучени са системи за …

Проектирана е система за …

Реализирана е система …

Приноси(научни/научно-приложни/приложни)

Научни - свравнителен анализ на съществуващи системи

Научно-приложни - проектиране

Приложни - имплементацията

Апробация - системата ще се ползва за …

Насоки за бъдеща работа, перспективи

Използвана литература

Google corp/org name. “Google title.” *Google website title*, 14 March 2022, https://google.com. Accessed 14 March 2022.