



УНИВЕРСИТЕТ ИТМО

Университет ИТМО

Кафедра компьютерных образовательных технологий

Разработка информационной системы создания,
сдачи и проверки заданий
по дисциплине "Программирование"

Автор: Цибин А.И.

Научный руководитель: Ефимчик Е.А., к.т.н., доцент кафедры КОТ

Санкт-Петербург, 2018

Актуальность

- Не являются автоматизированными задачи оценки качества кода работ и поиска плагиата в работах обучающихся.
- Существует ряд искусственных ограничений, накладываемых на архитектуру проверяемых решений.

Обзор аналогов

Система	Автоматизация проверок	Добавления собственных проверок решений	Возможность вывода дополнительной информации об ошибках
Coursera	Реализуется непосредственно автором		
Edx			
Hackerrank	Присутствует для тестовых пар вход-выход	Отсутствует	Присутствует отчет по парам вход-выход
Stepik		Реализуется непосредственно автором	
Cyber-Net			
MCE			

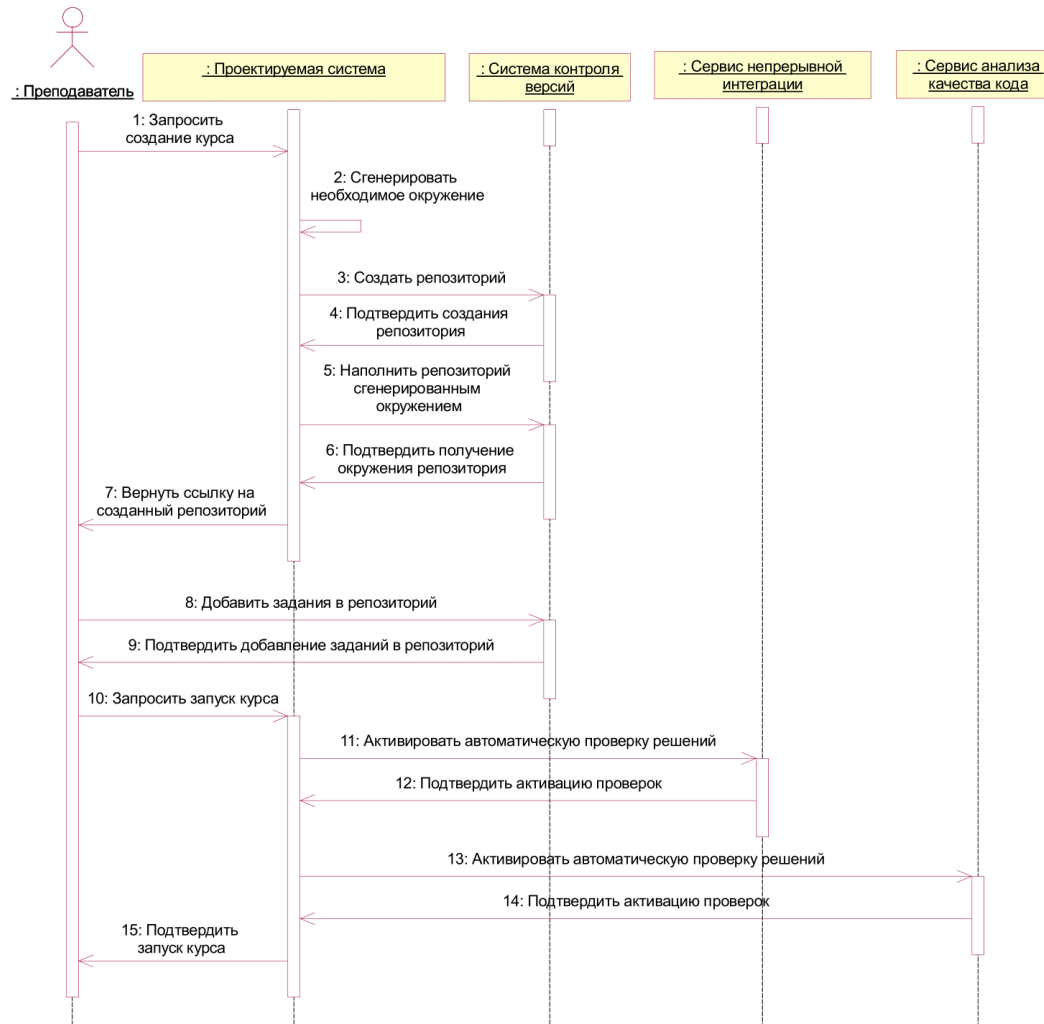
Выработанные требования

- Обучающийся должен иметь возможность быстро проверить свое решение.
- Спектр вариантов решения должен ограничиваться технически минимально.
- Всевозможные проверки решений должны проводиться автоматически.
- Задания должны быть организованы в полноценные образовательные курсы.
- Возможность генерирования инфраструктуры курса.
- Возможность экспорта статистики об успеваемости.

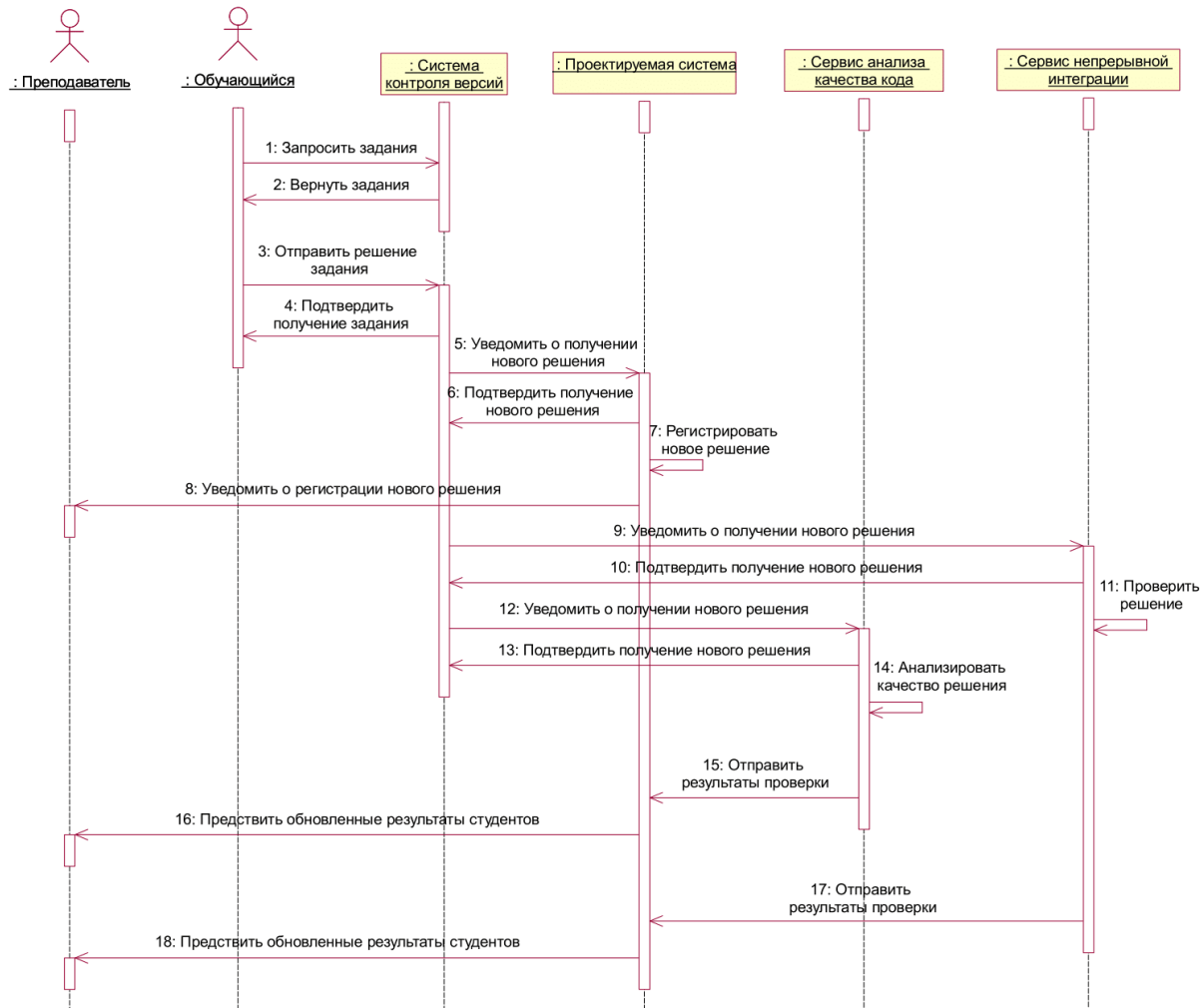
Решение

- Размещение заданий и решений в репозиториях системы контроля версий.
- Интеграция с сервисами:
 - непрерывной интеграции,
 - статического анализа кода,
 - поиска плагиата.

Механизм инициализации курса



Механизм сдачи и проверки



Механизм анализа плагиата

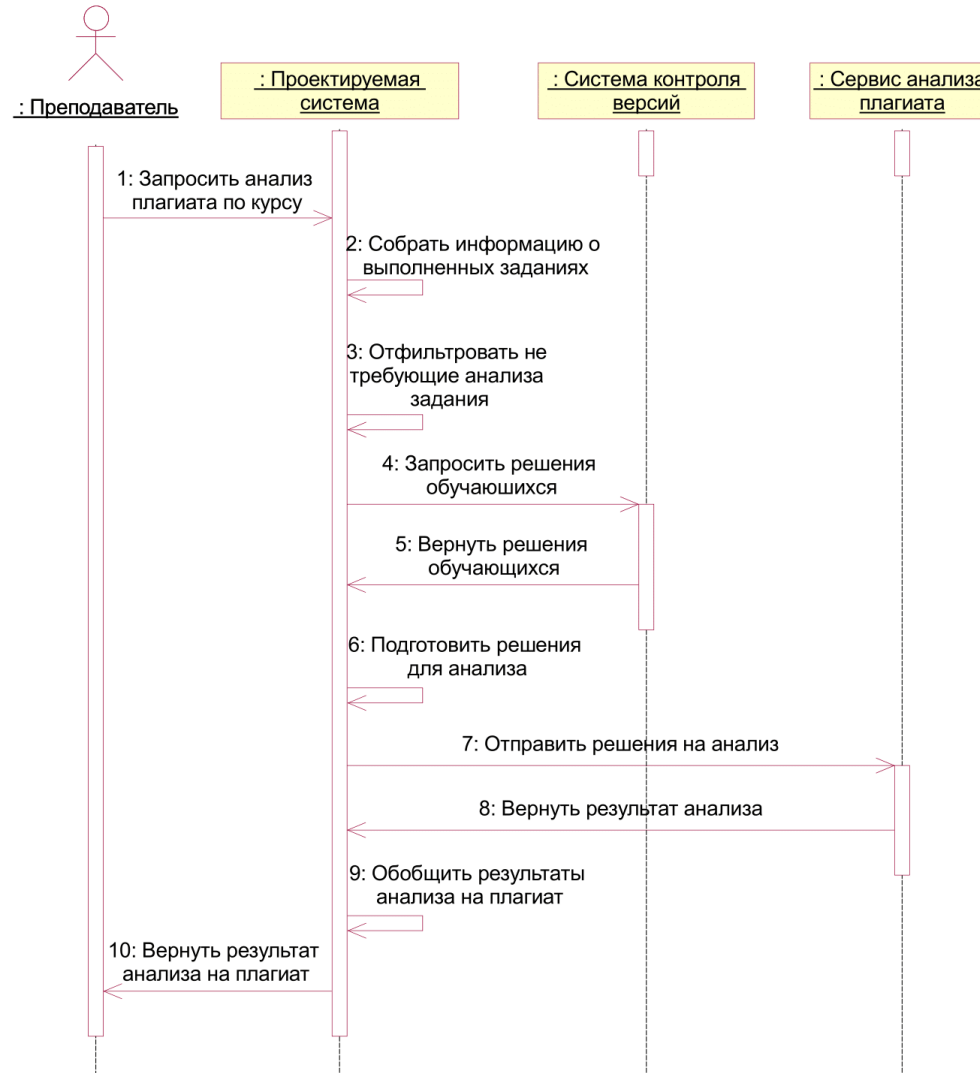
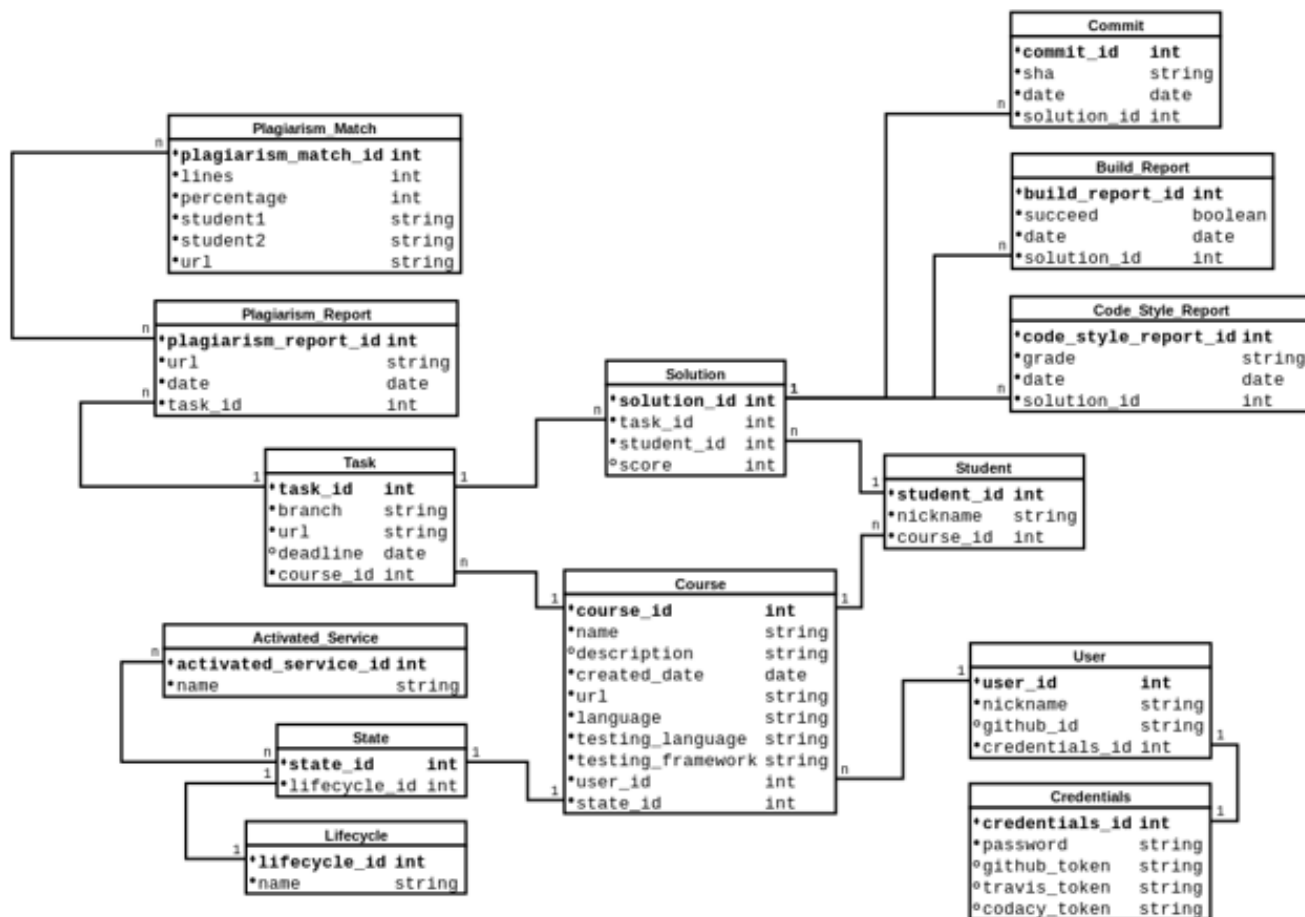


Схема база данных



Используемые инструменты

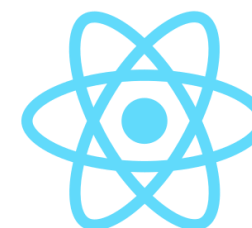
- Сборка
 - Gradle
- Сервер
 - Kotlin
 - Spring
 - Spek
- Клиент
 - JavaScript
 - Webpack
 - React



Spek



Gradle



* Полный список технологий и инструментов описан в репозитории проекта.

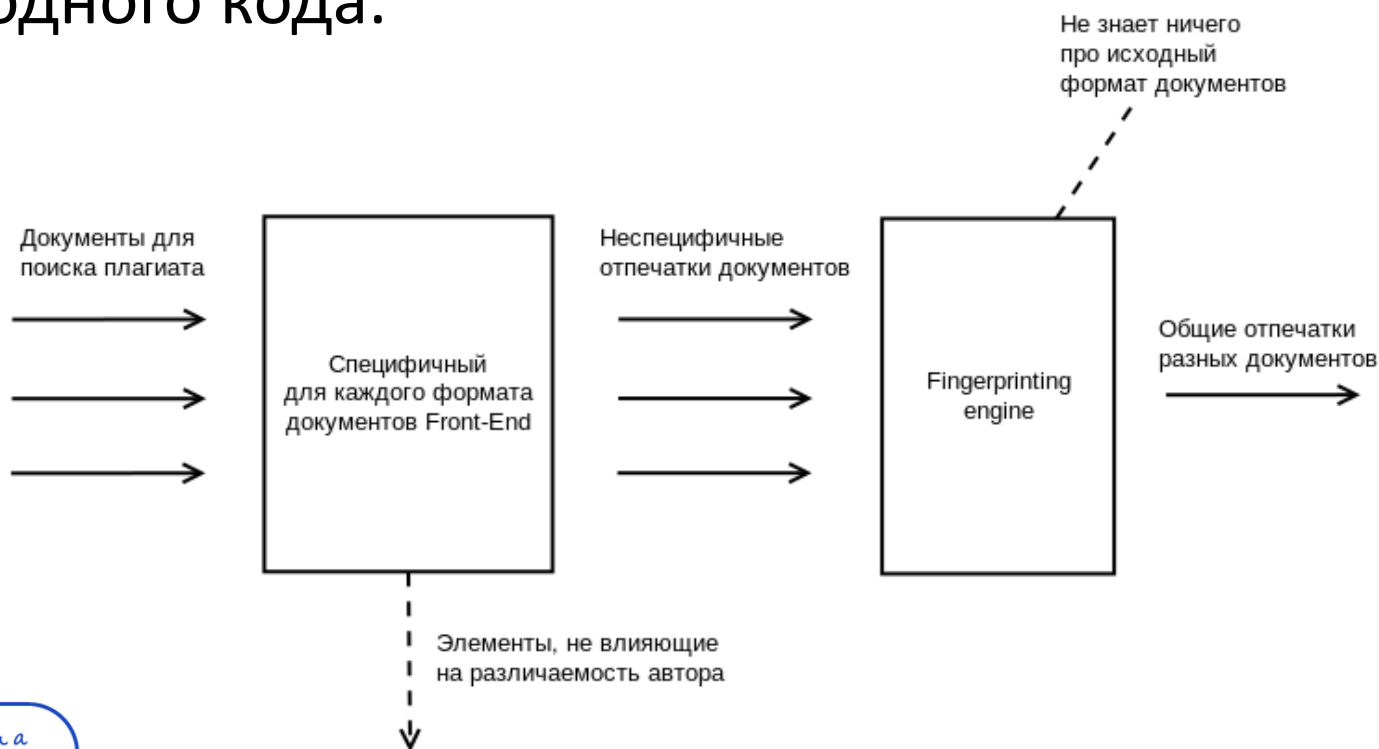
Технологии и вендоры

- Система контроля версий – GitHub (git):
 - знакомая для преподавателя система,
 - множество возможностей для разработчиков открытого программного обеспечения.
- Непрерывная интеграция – Travis:
 - подробные и при этом емкие отчеты,
 - широкие возможности кастомизации проверок,
 - интеграция с github.
- Статический анализ кода – Codacy:
 - большого количество инструментов статического анализа код,
 - интеграция с github.



Анализ плагиата

Moss (for a Measure Of Software Similarity) – автоматическая система определения схожести исходного кода.



Разработанная система

Flaxo

Signed as admin Options ▾

JavaCourse running travis codacy java java junit

Start course

Analyse plagiarism

Delete course

Download as ▾

Course summary

task1

task2

task1

Git branch

Plagiarism report

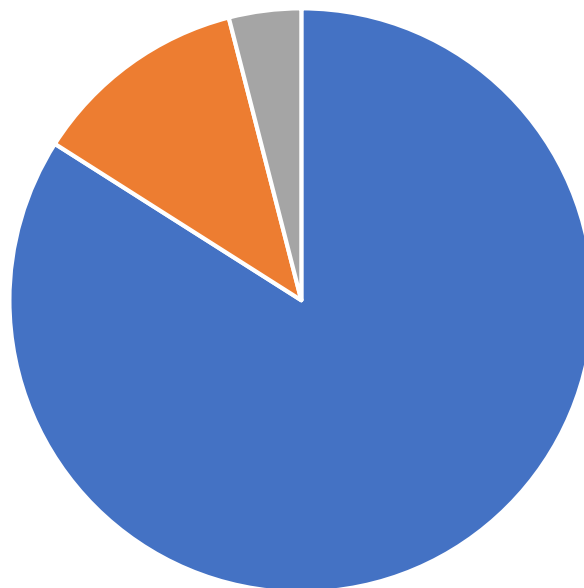
Save results

Rules

#	Student	Build	Code style	Plagiarism detection	Deadline	Result
1	student1	✓	B	1 85%	95	suggesting 95
2	student2	✓	A	1 22%	100	suggesting 100
3	student3	✗	B		35	suggesting 35
4	student4				0	suggesting 0
5	student5	✓	C	2 85%, 22%	80	suggesting 80

Внедрение

- Более 30 студентов
- Более 120 проверенных работ
- 4 замеченных случая плагиата



Общее впечатление
от курса

25 опрошенных

- понравилось
- все равно
- не понравилось

Заключение

Создан сервис для организации, проведения и проверки испытаний по программированию с функциями автоматизированной проверки решений, анализа качества кода и поиска случаев плагиата.

Исходный код и документация:
<https://github.com/tcibinan/flaxo>

Выступления на конференциях

Конгресс Молодых Ученых 2018, доклад по теме:
«Распределенная интегрированная
информационная система организации сдачи и
проверки заданий по программированию:
применение и особенности технической
реализации».



Спасибо за внимание!

Контакты:

tsibin.andr@gmail.com

[@tcibinan](https://github.com/tcibinan)

IT's *MO*re than a
UNIVERSITY