Описание на проект:

„Екосистема за оценяване на задачи по информатика“

Съдържание

[Въведение 2](#_Toc349846658)

[Описание на системата 2](#_Toc349846659)

[Критерии във връзка с програма ЕОС 2012-2013 4](#_Toc349846660)

[Необходимост от осъществяване на проекта 4](#_Toc349846661)

[Цел на проекта 5](#_Toc349846662)

[Целева група 6](#_Toc349846663)

[Същност на проектното предложение и планирани средства за достъп до резултатите от проекта 6](#_Toc349846664)

[Методология за реализацията на проекта 7](#_Toc349846665)

[Необходима инфраструктура за реализацията на проекта и за експлоатация на получените резултати 7](#_Toc349846666)

[Работна програма 7](#_Toc349846667)

[Ефект от изпълнението на проекта 8](#_Toc349846668)

# Въведение

Целта на този документ е да представи софтуерна система за целите на обучението по информатика, както на университетско, така и на училищно ниво. Системата е тясно интегрирана с вече налични системи, които се използват в учебния процес на Факултета по Математика и Информатика на Софийски университет „Св. Кл. Охридски“, но не зависи от тяхното наличие, за да функционира коректно.

# Описание на системата

Следната диаграма представя изглед от високо ниво на системата:



Системата се състои от следните компоненти и подсистеми:

**Arena Maycamp** и **Spoj0** –системи за оценяване на задачи, които се използват в курсовете по Дизайн и Анализ на Алгоритми и Състезателно програмиране. Следва да бъдат разширени с интерфейс от уеб услуги, които предоставят възможност за изпращане на задача за оценяване (с връщане на резултата), CRUD (create-read-update-delete) операции за състезания и задачи и извличане на статистика за задачи и състезания.

**MongoDB** – нерелационна база данни, която предоставя информацията унифицирано под формата на JSON формат (BSON на файловата системата - компилиран JSON). Изборът точно на MongoDB не е случаен и се дължи на следните фактори:

* вече наличен труд за описанието на общ формат и хранилище на задачи от състезателен характер е наличен под формата на дипломна работа във ФМИ, които представя хранилището под формата на файлове система, а описанието на данните – в JSON формат. Същата тази структура е реализирана и от MongoDB в много по-компактен формат. MongoDB освен това предоставя много повече възможности за скалируемост на системата, излишък на данните с цел по-голяма надежност, механизми за агрегация и map reduce върху данните, сигурност и редица допълнителни функционалности типични за една база от данни. Тези функционалности не са налични в описаното хранилище (CORE – Content Repository) за съхранение на задачите;
* лесната разработка и администрация на приложения, които използват MongoDB
* високата скорост на изпълнение на заявки и обработка на данните за разлика от една типична RDBMS

MongoDB ще се използва както за хранилището за задачи (Content Repository), така и за хранилището за резултатите и решенията към съответните задачи (Submission Repository).

**Advanced Grader** – система за оценяване, която дава възможност за оценка на задачи по следните критерии:

* eзик за програмиране – C/C++, Java, C#, Python
* оценявана единици – цяла програма (единичен файл), клас, функция или фрагмент от код спрямо съответния език за програмиране

Системата за оценяване позволява клъстеризация, което дава възможност за скалируемост.

**Grading Façade** – компонент, който позволява по унифициран начин да се изпрати задача за оценяване към някоя от посочените подсистеми, както и да се извлекат резултати и статистика за оценки на задачите от съответните системи за оценяване. Достъпът до съответните системи става постредством интерфейс от уеб услуги;

**Grading Unit Façade** – компонент, който позволява по унифициран начин да се извършват CRUD (create-read-update-delete) операции върху данните за съответните системи за оценяване. За системите spoj0 и maycamp модела се запазва, като манипулацията става посредством интерфейсите с уеб услуги (REST-базиран). Допълнително данните се пишат и в унифицирания формат в MongoDB. Този компонент позволява и конвертиране на данните (1-to-1) между съответните системи с цел максимална преизползваемост и трансфер на данни (задачи, състезания). Такава конверсия ще позволява вече съществуващи задачи и състезания от spoj0 и maycamp да преминат в унифицирания формат и обратното – от унифицирания формат да се създават задачи и състезания в съотвения модел на вече съществуващите системи.

**Client Façade** – компонент, който позволява на различни типове клиенти да използват централизирано услугите предоставяни от Grading Façade и Grading Unit Façade компонента

Към системата са налични и четири клиента – един за преподаватели и три за студенти/ученици:

**Admin interface** – административен интерфейс, който дава възможност да се добавят максимално лесно и бързо нови данни, в набор от наличните системи за оценяване. Допълнителни средства, който подпомагат генерирането на тестови данни и извличането на статистики от съотвените системи за оценяване също са налични.

**UI interface** – web-базиран графичен интерфейс за използване на системата и изпращане на код за оценка

**Eclipse plug-in** – разширение за средата Еклипс, което предоставя удобен начин за преглед на наличните състезания/задачи и оценяването на задача

**Visual Studio Add-on** – разширение за средата Visual Studio, което предоставя удобен начин за преглед на наличните състезания/задачи и оценяването на задача

# Критерии във връзка с програма ЕОС 2012-2013

**(разширен вариант, който обхваща всички компоненти на системата)**

## Необходимост от осъществяване на проекта

Практическите упражнения в обучението по информатика са незаменимо средство за придобиване на реален опит и по-добро усвояване на теоретичния материал. Текущия набор от средства, които се използват за тази цел в учебния процес по информатика във ФМИ (по-специално системите Arena Maycamp и Spoj0) налагат някои ограничения най-вече по отношение на ефективността и възможността за добавяне на съдържание:

* трудно добавяне на нови данни в системите – добавянето на данни (състезания, задачи) в текущите системи е с доста ограничена функционалност, която не позволяването да се правят модификации, а дори и самото добавяне е трудно и налага знание за спецификите на съответната използвана система;
* липса на общ формат за представяне на данните – липсва установен формат за представяне на състезанията, задачите и другите необходими данни за целите на практическата подготовка, което допълнително намалява и възможността за преизползване на тези данни;
* липса на интеграция със среди за разработка – при реализацията на софтуерни системи се използва най-вече среда за разработка, но текущите системи не предоставят такива средства (и по специално разширения за двете най-ширико разпространени среди – Eclipse и Visual Studio). Текущите системи предлагат възможност за качване на задачите и проверка на статус (брой успешни/неуспешни тестове) през външен потребителски интерфейс;
* липса на различни модели за предаване на задачите – текущите системи позволяват единствено оценяването на цели програми, но не и на отделни програмни фрагменти като например класове, функции и блокове от код;
* помощни средстава за генерирането на тестови данни за системите – в общия случай генерирането на тестови данни може да се окаже времеемък процес, който изисква написването на генератори на данните за съответни задачи. Един добър набор от генератори на тестови данни към системата за основни типове задачи би улеснил работата на предподавателя.

## Цел на проекта

Основната цел на проекта е реализацията на софтуерна система с пет основни направления:

* разширяване на възможностите на текущите системи използвани в обучението по информатика и интеграцията подсредством тези разширения – целта е да се използват вече натрупани данни;
* разработка на централизиран административен интерфейс за преподаватели, който да предоставя средства за максимални ефективно създаване на нови задачи – целта е да се улесни работата на предподавателите отговорни за създаването на данни за задачите и следене на прогреса по съответните задачи;
* интеграция с най-използваните среди за разработка и централизиран клиентски Web интерфейс за използване на системата – целта е да се подобри фокуса на учениците/студентите върху използването на конкретната среда и решаването на съответните задачи, както и да се минимизира времето за предаване и проверка на резултатите по съответните задачи;
* реализация на изцяло нова система за оценка на задачи, която разширява възможностите на вече текущите системи по отношение на формата на предаване на кода за оценка (цяла прорграма, клас, метод, фрагмент от код), както и на допустимите езици за реализация на задачите (C\C++, Java, C#, Python);
* унифицирано представяне на данните в системата (задачи, състезание и всякаква допълнителни информация) – целта е да се избегне двойната работа при създаването на нова задача, която може да се използва в повече от една система за оценяване.

Крайният резултат от изпълнението на посочените направления е напълно функционираща система за обучение с инсталация върху наличната инфраструктура във Факултета по Математика и Информатика на СУ. Като резултат от изпълнението на проекта следва да бъдат реализирани следните дейности:

* запознаване в дълбочина с реализацията и възможностите на вече налични системи за оценяване на задачи по информатика, които се използват в практиката, както и с теоретични разработки в насока системи за оценяване на задачи по информатика;
* запознаване в дълбочина с използваните технологиите и при необходимост обмен на знания и опит между участниците в проекта;
* документиран дизайн на отделните модули в системата и формална спецификация на външните интерфейси (уеб услуги) към всеки един от модулите на системата;
* реализация на системата под формата на подходящи проекти, които обхващат отделните модули на системата;
* клиентска документация за потребителите на системата;
* документация за инсталацията и администрацията на системата;
* документация на използваната инфраструктура в рамките на Факултета по Математика и Информатика;
* инсталацията на системата поне в рамките на Факултета по Математика и Информатика на СУ;
* инсталация на системата в поне едно профилирано учебно заведение с преподаване на информатика;
* информационен уеб портал на системата.

## Целева група

Потребителите на системата следва да бъдат преподавателите и студентите от специалностите по информатика, както в Софийския университет (в частност Факултета по Математика и Информатика – най-вече специалности Компютърни Науки, Информатика и Приложна математика), така и в други висши учебни заведения, които предлагат обучение по информатика. Системата ще бъде достъпна също така за преподаватели и ученици от учебните заведения. Системата може да се използва в курсовете по Увод в програмирането, Структури от Данни, Обектно-ориентирано програмиране и Дизайн и Анализ на Алгоритми, както и в редица по-специализирани курсове по програмиране.

## Същност на проектното предложение и планирани средства за достъп до резултатите от проекта

(описанието на системата в началото на документа – частта от нея, която следва да се реализира)

Необходимите файлове за инсталация на системата ще бъдат налични през уеб портала на системата. Сорс кода, дизайн документацията и останалите ресурси на системата ще бъдат налични в специално-предназначено за целта GIT хранилище.

## Методология за реализацията на проекта

Проектът следва да се разработи на базата на Waterfall модела с елементи на Agile (по-специално екстремно програмиране) поради две основни причини:

* кратки срокове за изпълнение
* възможности за надграждане на системата с допълнителни функционалности дори след завършване на първоначалната фаза на проекта.

## Необходима инфраструктура за реализацията на проекта и за експлоатация на получените резултати

Необходимата инфраструктура за инсталация на системата е вече налична във Факултета по Математика и Информатика на СУ. Участниците следва да използват собствена инфраструктура за тестова среда или при необходимост - допълнителни хостинг услуги.

## Работна програма

Първоначалната фаза на проекта включва следните етапи:

* **етап 1** – архитектура на системата, която включва: описание на разширенията към системите Arena Maycamp и spoj0, дизайн на фасадата за CRUD (create-read-update-delete) операции към системите за оценяване, дизайн на администравния интерфейс BRAIN, схема на хранилището за задачи и допълнителни изгледи на системата. Този етап включва и допълнително запознаване с използваните технологии и библиотеки (при необходимост), както и споделяне на опит и знания между участниците. Срок за изпълнение: **10.03.2013 – 10.04.2013;**
* **етап 2** – реализация и тестване на системата, която включва цялостното реализиране на системата под формата на сорс код и допълнителните ресурси към него. Срок за изпълнение: **10.04.2013 – 10.08.2013;**
* **етап 3** – техническа документация за потребителите и администраторите на системата, която включва документ за използване, документ за инсталация и документ за администрация. Срок за изпълнение: **10.08.2013 - 10.09.2013;**
* **Етап 4** – инсталация на системата на наличната инфраструктура във Факултета по Математика и Информатика на СУ. **етап 3** – техническа документация за потребителите и администраторите на системата, която включва документ за използване, документ за инсталация и документ за администрация. Срок за изпълнение: **10.08.2013 - 10.09.2013;**

(Забележка: етап 3 и етап 4 следва да се реализират паралелно).

Участниците в проекта следва да участват във всеки един етап от реализацията му.

## Ефект от изпълнението на проекта

(описанието на системата в началото на документа – частта от нея, която следва да се реализира на следващ етап)

Реализацията на първоначалния етап на системата следва да повиши значително ефективността от използване на вече наличните системи за оценяване на задачи, което ще улесни значително процеса по организация и ръководене на съотвените курсове, които в момента използват системата. Също така първоначалния етап ще постави основите за реализацията на редица допълнителни компоненти към системата.

Резултатите от изпълнението на проекта ще бъдат налични в публично **github** хранилище на следния адрес: <https://github.com/martinfmi/grading_ecosystem>