|  |  |
| --- | --- |
| Министерство образования и науки Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  «Ижевский государственный технический университет  имени М. Т. Калашникова» | |
| Кафедра «Программное обеспечение» | |
| Отчёт по лабораторной работе №2  дисциплины «КПО» | |
| Выполнил:  Студент группы Б08-191-2 | Р. С. Поскребышев |
| Принял: | В.Г. Власов |
| Ижевск 2016 | |

1. Система

Распределенная система, ориентированная на управление задачами и проверку их решений в сфере олимпиадного программирования.

1. Подсистемы

Archive API

1. Archive API (Backend)
2. Archive API Spec (Frontend)
3. Подсистема репозиториев
4. Подсистема ролей
5. Подсистема авторизации
6. Подсистема управления архивом

Executor API

1. Очередь решений
2. Executor API (Backend)
3. Executor API Spec (Frontend)
4. Подсистема репозиториев
5. Подсистема ролей
6. Подсистема авторизации
7. Подсистема тестирования задач
8. Модули

Archive API

1. Archive API (Backend)
   1. Модуль получения множества задач
   2. Модуль получения ссылки на условие задачи
2. Archive API Spec (Frontend)
   1. Модуль страницы спецификации
3. Подсистема репозиториев
   1. Модуль добавления записи
   2. Модуль получения записи
   3. Модуль обновления записи
   4. Модуль удаления записи
4. Подсистема ролей
   1. Модуль установки ограничения на выполнение метода по ролям
5. Подсистема авторизации
   1. Модуль проверки правильности и существования токена
   2. Модуль получения данных из токена
6. Подсистема управления архивом
   1. Модуль получения множества задач
   2. Модуль получения ссылки на условие

Executor API

1. Очередь решений
   1. Модуль добавления решения в очередь
   2. Модуль получения результата из очереди
2. Executor API (Backend)
   1. Модуль отправки множества решений на проверку
   2. Модуль перепроверки множества решений
   3. Модуль получения множества результатов
   4. Модуль получения списка поддерживаемых языков программирования
3. Executor API Spec (Frontend)
   1. Модуль страницы спецификации
4. Подсистема репозиториев
   1. Модуль добавления записи
   2. Модуль получения записи
   3. Модуль обновления записи
   4. Модуль удаления записи
5. Подсистема ролей
   1. Модуль установки ограничения на выполнение метода по ролям
6. Подсистема авторизации
   1. Модуль проверки правильности и существования токена
   2. Модуль получения данных из токена
7. Подсистема тестирования задач
   1. Модуль отправки множества решений на проверку
   2. Модуль перепроверки множества решений
   3. Модуль получения множества результатов
   4. Модуль получения списка поддерживаемых языков программирования
8. Диаграмма модулей



1. Вопросы
2. Нет стартового модуля Archive API (Backend);
3. Нет стартового модуля Executor API (Backend);
4. Разделить подсистемы репозиториев для Archive и Executor.
5. Исправления

Archive API

1. Archive API (Backend)
   1. Модуль получения множества задач
   2. Модуль получения ссылки на условие задачи
   3. Модуль страта
2. Archive API Spec (Frontend)
   1. Модуль страницы спецификации
3. Подсистема репозиториев
   1. Модуль добавления записи
   2. Модуль получения записи
   3. Модуль обновления записи
   4. Модуль удаления записи
4. Подсистема ролей
   1. Модуль установки ограничения на выполнение метода по ролям
5. Подсистема авторизации
   1. Модуль проверки правильности и существования токена
   2. Модуль получения данных из токена
6. Подсистема управления архивом
   1. Модуль получения множества задач
   2. Модуль получения ссылки на условие

Executor API

1. Очередь решений
   1. Модуль добавления решения в очередь
   2. Модуль получения результата из очереди
2. Executor API (Backend)
   1. Модуль отправки множества решений на проверку
   2. Модуль перепроверки множества решений
   3. Модуль получения множества результатов
   4. Модуль получения списка поддерживаемых языков программирования
   5. Модуль старта
3. Executor API Spec (Frontend)
   1. Модуль страницы спецификации
4. Подсистема репозиториев
   1. Модуль добавления записи
   2. Модуль получения записи
   3. Модуль обновления записи
   4. Модуль удаления записи
5. Подсистема ролей
   1. Модуль установки ограничения на выполнение метода по ролям
6. Подсистема авторизации
   1. Модуль проверки правильности и существования токена
   2. Модуль получения данных из токена
7. Подсистема тестирования задач
   1. Модуль отправки множества решений на проверку
   2. Модуль перепроверки множества решений
   3. Модуль получения множества результатов
   4. Модуль получения списка поддерживаемых языков программирования



1. Интерфейсы

IArchiveManager

public interface IArchiveManager

{

ICollection<Problem> GetCollection(long collectionId, string token);

string GetStatementUrl(long problemId, string token);

}

IArchiveWebSpec

public interface IArchiveWebSpec

{

}

IAuthManager

public interface IAuthManager

{

bool CheckToken(string token);

Token GetData(string token);

}

IBaseRepository

public interface IBaseRepository

{

object Get(long id);

object Add(object entry);

void Update(object entry);

void Delete(object entry);

}

IExecutorWebSpec

public interface IExecutorWebSpec

{

}

ILangRepository

public interface ILangRepository : IBaseRepository

{

}

ILangsController

public interface ILangsController

{

ILangRepository LangRepository { get; }

ITestingManager TestingManager { get; }

ICollection<Language> GetLangs();

}

IProblemRepository

public interface IProblemRepository : IBaseRepository

{

}

IProblemsController

public interface IProblemsController

{

IProblemRepository ProblemRepository { get; }

IAuthManager AuthManager { get; }

IArchiveManager ArchiveManager { get; }

ICollection<Problem> GetCollection(long collectionId, string token);

string GetStatementUrl(long problemId, string token);

}

IRabbitQueue

public interface IRabbitQueue

{

void Add(Submit submit);

SubmitResult GetSubmitResult(long submitId);

}

IResultRepository

public interface IResultRepository : IBaseRepository

{

}

IResultsController

public interface IResultsController

{

IResultRepository ResultRepository { get; }

IAuthManager AuthManager { get; }

ITestingManager TestingManager { get; }

ICollection<SubmitResult> GetSubmitResults(ICollection<long> ids, string token);

}

IRoleManager

public interface IRoleManager

{

void SetAccess(string methodId, string token);

}

ISubmitRepository

public interface ISubmitRepository : IBaseRepository

{

}

ISubmitsController

public interface ISubmitsController

{

IRabbitQueue RabbitQueue { get; }

ISubmitRepository SubmitRepository { get; }

IAuthManager AuthManager { get; }

ITestingManager TestingManager { get; }

ICollection<long> Send(ICollection<Submit> submits, string token);

void Rejudge(ICollection<long> submitIds, string token);

}

ITestingManager

public interface ITestingManager

{

ICollection<long> Send(ICollection<Submit> submits);

void Rejudge(ICollection<long> submitIds);

ICollection<SubmitResult> GetSubmitResults(ICollection<long> ids);

ICollection<Language> GetLangs();

}

Executor::IStartup

public interface IStartup

{

IExecutorWebSpec WebSpec { get; }

ILangsController LangController { get; }

IResultsController ResultsController { get; }

ISubmitsController SubmitsController { get; }

}

Archive::IStartup

public interface IStartup

{

IArchiveWebSpec WebSpec { get; }

IProblemsController ProblemsController { get; }

}

ПРИЛОЖЕНИЕ

Техническое задание к проекту

# Введение

# Назначение системы

Система предназначена для управления задачами, управления доступом, тестирования решений.

# Область применения системы

Система ориентирована на учебные заведения (школы, университеты и др.) и иные учреждения, где есть необходимость автоматизированной проверки решений.

# Определения, акронимы, аббревиатуры

Тестирование, проверка, хранилище задач.

# Обзор системы

Система представляет собой комбинацию из двух основных приложений: программный интерфейс для работы с архивом, программный интерфейс для работы с тестирующей системой, и 2 графических интерфейса. Графический интерфейс содержит спецификацию (описание методов API и моделей). Для выполнения операций необходим токен, выданный CAA, который идентифицирует пользователя.

# Общее описание системы

# Режимы и состояния системы

Веб-интерфейс работает все время и не имеет состояний. API также не хранят состояния пользователя.

# Основные функциональные возможности системы



# Основные условия системы

Для работы системы необходим сервер Windows.

# Оперативные сценарии

Система не предполагает дополнительных сценариев, работает в автономном режиме.