МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ В.Ф.
УТКИНА»

Кафедра «Вычислительная и прикладная математика»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Распределенные системы обработки информации»

Направление подготовки – 09.04.04 Программная инженерия

ОПОП академической магистратуры

«Программно-алгоритмическое обеспечение систем искусственного интеллекта»

Квалификация (степень) выпускника – магистр

Форма обучения – очная (2 года)

Рязань

1.ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Перечень компетенций (планируемых результатов освоения образовательной программы), выявленных в матрице компетенций, представлен в таблице 1 рабочей программы дисциплины совместно с планируемыми результатами обучения по дисциплине, а также в таблице 1 фонда оценочных средств (раздел 2) с указанием этапов (семестров) их освоения.
Результаты обучения вносят свой вклад в формирование различных компетенций, предусмотренных образовательной программой. В свою очередь, компетенции на разных уровнях категорий «знать», «уметь», «владеть» формируются модулями (разделами) дисциплины, а также различными дисциплинами образовательной программы.

2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Фонд оценочных средств (ФОС) предусматривает:

описание комплекса **показателей** – дескрипторов освоения компетенций в виде результатов обучения, которые студент может продемонстрировать (таблица 1). Для контроля достижения каждого из них предусмотрены оценочные средства в виде вопросов, заданий и т.д.;

обозначение **критериев** – правил принятия решения по оценке достигнутых результатов обучения и сформированности компетенций.

Показатели достижения планируемых результатов обучения и критерии их оценивания на разных уровнях формирования компетенций приведены в таблице 1.

Таблица 1. Показатели достижения индикаторов компетенции

1	2	3	4
Компетенция: код по ФГОС 3++, формулировка	**Индикаторы**	**Этап**	**Наименование оценочного средства**
ОПК-8 (09.04.04) Способен осуществлять эффективное управление разработкой программных средств и проектов	ЗНАТЬ - методы управления разработкой программных средств и проектов	1, 2	Рубежные контроли. Экзамен. Курсовая работа
ПК-4 (09.04.04/11 Программное обеспечение систем искусственного интеллекта) Способен управлять проектами по созданию, поддержке и использованию систем бизнес-аналитики в организации	ЗНАТЬ - задачи и роль систем бизнес-аналитики в поддержке принятия решений в процессе управления организацией, принципы построения систем бизнес-аналитики - методы, технологии, инструменты и платформы бизнес-аналитики - методы анализа данных, используемых в системах бизнес-аналитики для принятия решений - методологию и принципы руководства проектом по созданию, поддержке и использованию систем бизнес-аналитики в организации УМЕТЬ - моделировать и анализировать процессы принятия управленческих решений и разрабатывать требования к системам бизнес-анализа в различных сферах деятельности - применять методы, инструменты и цифровые платформы анализа данных при проектировании и	1, 2	Рубежные контроли. Экзамен. Курсовая работа

1	2	3	4
	построении систем бизнес-аналитики - решать задачи по руководству коллективной проектной деятельности для создания, поддержки и использования системы бизнес-аналитики - оценивать результаты внедрения системы бизнес-аналитики в организации и разрабатывать рекомендации по совершенствованию и развитию системы		
ПК-8 (09.04.04/11 Программное обеспечение систем искусственного интеллекта) Способен осуществлять руководство по созданию и развитию систем и комплексов обработки данных, в том числе больших данных, для корпоративных и государственных заказчиков	ЗНАТЬ - принципы разработки проектной документации центра обработки данных - системную архитектуру серверного оборудования и систем хранения данных, цифровых платформ анализа данных - принципы и методы построения общедоступных платформ для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных - принципы и методы хранения наборов данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных систем видеонаблюдения) на общедоступных платформах для	1, 2	Рубежные контроли. Экзамен. Курсовая работа

1	2	3	4
обеспечения потребностей организаций-разработчиков в области искусственного интеллекта УМЕТЬ	существующие и перспективные структуры центров обработки данных действующую нормативную базу в области проектирования и строительства центров обработки данных формулировать технические задания по формированию аппаратного обеспечения и программных комплексов центра обработки данных вести переговоры с подрядчиками и поставщиками оборудования и программного обеспечения для центра обработки данных ставить задачи, планировать и контролировать выполнение работ сотрудниками центра обработки данных работать с UNIX-подобными системами применять принципы и методы построения общедоступных платформ для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных применять принципы и методы		

1	2	3	4
хранения наборов данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных систем видеонаблюдения) на общедоступных платформах для обеспечения потребностей организаций- разработчиков в области искусственного интеллекта			

ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИЕ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

ФОС по дисциплине содержит следующие оценочные средства, позволяющие оценить знания, умения и уровень приобретенных компетенций при текущем контроле и промежуточной аттестации, разбитые по модулям дисциплины:

примеры типовых вопросов для защиты лабораторных работ;

перечень вопросов к экзамену и макет экзаменационного билета;

комплекты заданий рубежных контролей;

макет типового задания на курсовой работы и перечень типовых вопросов для защиты.

Средства для оценки различных уровней формирования компетенций по категориям «знать», «уметь», «владеть» обеспечивают реализацию основных принципов контроля, таких, как объективность и независимость, практико-ориентированность, междисциплинарность.

учетом этого, контрольные вопросы (задания, задачи,) входящие в ФОС, для различных категорий и уровней освоения компетенций имеют следующий вид:

Уровень ЗНАТЬ

Дескрипторы	**Пример задания из оценочного средства**
методы управления разработкой программных средств и проектов	РСОИ и какие задачи они решают? Определение РСОИ и следствия из определения. Требования ко РСОИ.
задачи и роль систем бизнес-аналитики в поддержке принятия решений в процессе управления организацией, принципы построения систем бизнес-аналитики	РСОИ и какие задачи они решают? Определение РСОИ и следствия из определения.
методы, технологии, инструменты и платформы бизнес-аналитики	Придумать предметную область и реализовать 3-4 сервиса, выполняющих некоторую свою бизнес-функцию в рамках предметной области.
методы анализа данных, используемых в системах бизнес-аналитики для принятия решений	Что такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных.
методологию и принципы руководства проектом по созданию, поддержке и использованию систем бизнес-аналитики в организации	Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход.
существующие и перспективные структуры центров обработки данных	Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход. Микросервисная архитектура.
действующую нормативную базу в области	Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход.

проектирования и строительства центров обработки данных	Микросервисная архитектура.
принципы разработки проектной документации центра обработки данных	Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход. Микросервисная архитектура.
системную архитектуру серверного оборудования и систем хранения данных, цифровых платформ анализа данных	Хранение данных RAID. Репликация.
принципы и методы построения общедоступных платформ для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных	МоноЧто такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных. Хранение данных RAID. Репликация.
принципы и методы хранения наборов данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных систем видеонаблюдения) на общедоступных платформах для обеспечения потребностей организаций-разработчиков в области искусственного интеллекта	Что такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных.

Уровень УМЕТЬ

Дескрипторы	**Пример задания из оценочного средства**
моделировать и анализировать процессы принятия управленческих решений и разрабатывать требования к системам бизнес-анализа в различных сферах деятельности	Придумать предметную область и реализовать 3-4 сервиса, выполняющих некоторую свою бизнес-функцию в рамках предметной области.
применять методы, инструменты и цифровые платформы анализа данных при проектировании и построении систем бизнес-аналитики	Write Ahead Log. Связь с транзакциями.
решать задачи по руководству коллективной проектной деятельности для создания, поддержки и использования системы бизнес-аналитики	Требования ко РСОИ.
оценивать результаты	Требования ко РСОИ.

внедрения системы бизнес-аналитики в организации и разрабатывать рекомендации по совершенствованию и развитию системы	
формулировать технические задания по формированию аппаратного обеспечения и программных комплексов центра обработки данных	Хранение данных RAID. Репликация.
вести переговоры с подрядчиками и поставщиками оборудования и программного обеспечения для центра обработки данных	Хранение данных RAID. Репликация.
ставить задачи, планировать и контролировать выполнение работ сотрудниками центра обработки данных	Map-Reduce.
работать с UNIX-подобными системами	Проблема синхронизации в распределенных системах.
применять принципы и методы построения общедоступных платформ для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных	Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход.
применять принципы и методы хранения наборов данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных систем видеонаблюдения) на общедоступных платформах для обеспечения потребностей организаций- разработчиков в области искусственного интеллекта	Что такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных. Хранение данных RAID. Репликация.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

4.1. Примеры методических материалов, определяющих процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Рубежный контроль	Средство проверки освоения уровней «знать», «уметь» компетенций ФГОС 3++	Комплекты билетов рубежных контролей
Экзамен	Средство проверки освоения уровня «знать» компетенций ФГОС 3++	Перечень вопросов к экзамену и макет экзаменационного билета
Курсовой проект	Средство проверки освоения уровня «владеть» компетенций ФГОС 3++	Примерное задание на курсовой проект и перечень типовых вопросов для защиты

Комплект билетов к рубежному контролю № 1

Билет № 1

Написать простейшее веб-серверное приложение и реализовать его сборку, используя готовые решения, которые предоставляют Bitbucket и Gitlab.

Билет № 2

Реализовать интеграционное тестирование веб-серверного приложения.

Билет № 3

Написать простейшее веб-серверное приложение и реализовать сборки, unit-тестирования и деплоя на сервер.

Комплект билетов к рубежному контролю № 2

Билет № 1

Реализовать OAuth2 авторизацию для API и пользовательского интерфейса, также использование токена при взаимодействии между сервисами.

Билет № 2

Придумать предметную область и реализовать 3-4 сервиса, выполняющих некоторую свою бизнес-функцию в рамках предметной области.

Билет № 3

Реализовать три сервиса, выполняющих некоторую свою бизнес-функцию в рамках заданной предметной области.

Перечень вопросов к экзамену

РСОИ и какие задачи они решают? Определение РСОИ и следствия из определения.
Требования ко РСОИ.
Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход.
Микросервисная архитектура.
Протоколы сериализации.
Протокол HTTP. На каком уровне работает, какие плюсы и минусы. Методы HTTP, статусы и заголовки.
Сессия. Что такое Cookie и для чего они используются. Stateful и Stateless.
Угрозы в сети. DDoS, XSS, CSRF, Man-In-The-Middle. Что такое CORS?
HTTPS.
Авторизация и аутентификация. OAuth, OAuth 2, OpenID Connect.
Что такое Front-End и какие задачи он выполняет? Масштабирование Front-end'ов.
Виды кэширования (code, proxy, browser). Как работает кэш в браузере.
Что такое RESTful, принципы его использования.
Заявки, жизненный цикл заявки. Идентификаторы заявок. Обработка таймаутов.
Проблема синхронизации в распределенных системах.
Асинхронное взаимодействие сервисов.
Некорректное выполнение операций. Повторная попытка, отмена операций, XA-транзакции.
Теорема CAP.
Транзакции. ACID. Проблемы транзакций. Оптимистическая и пессимистическая блокировка.
Write Ahead Log. Связь с транзакциями.
Что такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных.
Что такое BASE. Что такое eventual consistency, почему этот подход применим.
Map-Reduce.
Хранение данных RAID. Репликация.
Шардирование.

Макет оформления экзаменационного билета

ФГБОУ ВО РГРТУ

Экзаменационный билет № 1

по курсу «Распределенные системы обработки информации»

РСОИ и какие задачи они решают?
баллов

Транзакции. ACID. Проблемы транзакций. Оптимистическая и пессимистическая блокировка.
баллов

Шардирование.
10 баллов

Билет рассмотрен и утвержден на заседании кафедры ИУ7 «___» _____ 20 __ г.

Перечень лабораторных работ

Лабораторные работы №1.1 - №1.4. Создание простейшего веб-серверного приложения

Цель работ: изучение и реализация простейшего веб-серверного приложения.

рамках первых 4 лабораторных работ требуется написать простейшее веб-серверное приложение и реализовать сборки, unit-тестирования и деплоя на сервер. Требования для выполнения работы:

Исходный проект хранится в удаленном репозитории (Github, Bitbucket, Gitlab).
Для сборки можно использовать Travic CI или воспользоваться готовыми решени-ями, которые предоставляют Bitbucket и Gitlab.
В build-плане сделать раздел сборки, тестирования и деплоя.
Для деплоя выбрать платфому PaaS типа Heroku, OpenShift или подобных. Если есть интерес, можно выбрать бесплатный VPS-сервер и натсроить деплой на него.
(опционально) Реализовать интеграционное тестирование.

Лабораторные работы №1.5 - №1.8. Реализация взаимодействия сервисов (12 час.)

Цель работ: изучение и реализация взаимодействия друг с другом сервисов.

рамках данных 4 лабораторных работ требуется выбрать предметную область и реализовать несколько взаимодействующих друг с другом сервисов. Сервисы должны ре-шать только свою бизнес-функцию, которая является частью общей системы. Система должна предоставлять REST API для выполнения операций. Получение данных нужно выполняться только через сервис, к которому относятся эти данные.

Агрегацию результатов выполнять на отдельном сервисе.
Требования для выполнения работы:

Придумать предметную область и реализовать 3-4 сервиса, выполняющих некоторую свою бизнес-функцию в рамках предметной области.
Помимо 3-4 сервисов выделить сервис агрегации (gateway), проксирующий запросы от других систем и агрегирующий данные. Все запросы пользователя проходят через эту систему.

Данные каждого сервиса можно хранить как в SQL, так и в NoSQL базе. Для упроще-ния допускается хранить данные на одной базе, но в разных схемах. При этом каждый сервис должен взаимодействовать только со своей схемой, получение данных, не от-носящихся к текущему сервису строго запрещено.
Должен быть хотя бы один запрос, требующий агрегированную информацию с не-скольких сервисов (двух и более).
Должно быть минимум два запроса, выполняющие обновление данных на нескольких сервисах в рамках одной операции.
При получении списка данных предусмотреть пагинацию.
Сделать подробное логгирование выполняемых операций на каждом сервисе.
При реализации использовать шаблоны MVC, смешивание представления и бизнес-логики строго запрещено.
Все публичные методы требуется покрыть unit-тестами (покрытие функций). Покры-тие тестами должно быть не менее 70%, для демонстрации покрытия использовать ка-кую-нибудь утилиту.
Подготовить шаблоны запросов или маленький скрипт для демонстрации работы.

Лабораторные работы№2.1 - №2.4 Реализация механизмов, увеличивающих отказо-устойчивость системы.

Цель работ: реализовать механизмы, увеличивающие отказоустойчивость систе-мы.

рамках данных 4 лабораторных работ требуется, используя ПО, разработанное ранее в лабораторных работах 1-8, реализовать механизмы, увеличивающие отказоустойчивость системы.

Требования для выполнения работы:

В случае недоступности части систем или внутренних ошибок сервера, возвращать ошибку с пояснением (т.е. http статус 4xx или 5xx + json с пояснением).
Реализовать валидацию входных данных. Если данные имеют некорректный фор-мат возвращать 400 Bad Request.
Для агрегирующего запроса на чтение (п. 4 требований ЛР 2) в случае недоступно-сти одной из систем, выполнять деградацию функциональности.
Для одного из запросов, выполняющих обновление данных на нескольких системах (п. 5 требований ЛР 2), в случае недоступности одной из систем, выполнять полный откат операции.
Для другого запроса, в случае недоступности одной из систем, возвращать пользо-вателю успешный результат, а операцию ставить в очередь для повторного выполнения. Когда система, недоступная в процессе операции, будет поднята, операция должна быть выполнена. Для очереди можно использовать in-memory очередь, например, Redis, ZeroMQ, или реализовать очередь на примитивах языка (BlockingQueue для Java).

Лабораторные работы №2.5 - №2.7 Реализация пользовательского интерфейса, выпол-няющего все операции, реализованные в REST API.

Цель работ: реализовать пользовательский интерфейс, выполняющий все опера-ции, реализованные в REST API.

рамках данных 3 лабораторных работ требуется, используя ПО, разработанное ранее в лабораторных работах 1-8, реализовать пользовательский интерфейс, выполняющий все операции, реализованные в REST API.

Требования для выполнения работы:

Для всех операций, представленных через REST API, написать вызов с пользова-тельского интерфейса.
В рамках лабораторной никакой авторизации не требуется. Если для выполнения операции требуется пользователь, то просто выбирать одного из заранее заданных.
Реализовать валидацию данных как на клиенте, так и на сервере.
Ошибки валидации требуется отображать на форме и помечать некорректные поля с кратким пояснением.
В случае неуспешного ответа от сервера, выводить пользователю уведомление на странице или всплывающим окном.

Лабораторная работа №2.8 Реализация OAuth2-авторизации для API и пользовательско-го интерфейса.

Цель работы: реализовать OAuth2 авторизацию для API и пользовательского ин-терфейса.

рамках данной лабораторной работы требуется, используя ПО, разработанное ранее в лаб. работах 1-8, реализовать OAuth2 авторизацию для API и пользовательского интер-фейса, а также использование токена при взаимодействии между сервисами. Требования для выполнения работы:

Все операции между сервером и клиентом разделить на публичные и закрытые. В REST API для выполнения закрытых операций использовать токен, полученный с помо-щью OAuth 2 (code flow).
Для пользовательского интерфейса реализовать login/password авторизацию, в ре-зультате на успешную авторизацию выдавать токен, идентифицирующий авторизованного пользователя. Все закрытые операции должны выполняться с помощью этого токена. Если для UI используется SPA и вся логика работы с сервером реализуется с помощью JavaScript, то токен передавать в заголовке Authorization: Bearer <token>. Если использует-ся server rendering и для отправки формы выполняется синхронный submit, то токен можно хранить в cookies и передавать через заголовок Cookie: <token-name>=<token>.
Регистрацию нового пользователя делать не обязательно, достаточно создать не-сколько тестовых пользователей.
Пароль в базе хранить в хэшированном виде.
Выделить отдельный сервис, отвечающий за авторизацию. Все запросы от UI и API на проверку токенов делегируютя ему через Aggregation Service.
Выданные токены хранить в некотором хранилище (можно в БД, redis, hazelcast и т.п.). На сервере не использовать встроенную сессию.
Для взаимодействия между сервисами использовать токен, каждый сервис имеет <appnId>/<appSecret>, по этим данным выдается токен с некоторым временем жизни. По окончанию времени жизни, токен требуется перезапросить. Авторизационные данные для доступа к каждому сервису знают сами сервисы, эти запросы не делегируются на Session Service.
Все токены должны иметь время жизни.
Для токена, полученного с помощью OAuth2, реализовать refresh токен.
Токен, используемый в пользовательском интерфейсе, должен инвалидироваться после 30 минут бездействия.
Подготовить пример получения/обновления OAuth 2 токена, выполнения запросов к REST API.
(опционально) Каждому сервису присвоить доменное имя (через /etc/hosts), между сервисами использовать HTTPS. Использовать nginx в качестве reverse-proxy, настройки HTTPS выполнять на нем.

Лабораторная работа №2.9 Реализация сервиса статистики.

Цель работы: реализовать сервис статистики.

рамках данной лабораторной работы требуется, используя ПО, разработанное ранее в лаб. работах 1-8, выделить сервис статистики, и отправлять на него информацию обо всех произошедших в системе действиях.

Требования для выполнения работы:

Выделить отдельный сервис, на который будут приходить информация с других сервисов через очередь (т.е. событийная модель). Среди этой информации должна быть оповещение об удачной/неудачной авторизации, запросах, изменяющих данные в системе и т.п.
Реализовать отказоустойчивую доставку событий (подтверждения успешной обра-ботки). При этом выполнять ожидание подтверждения обработки в другом потоке не бло-кируя сам запрос. Обрабатывать ситуации отсутствия ответа (таймаут ожидания) и ответ, что запрос не смог обработаться корректно из-за ошибок в данных. В случае если ответ не пришел, реализовать N раз повторную отправку. Если пришел ответ о некорректных дан-ных в сообщении, писать подробную информацию в лог.
Реализовать идемпотентную обработку событий на стороне приемщика.
Реализовать три сводных отчета по полученным данным. Реализовать графическое представление этих отчетов.

Добавить отдельную роль ADMIN, который будет иметь доступ к графическому представлению отчетов в специальной секции сайта, доступной только пользователям с этой ролью.

Макет типового задания на курсовой проект

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования РГРТУ

УТВЕРЖДАЮ
Заведующий кафедрой ВПМ
_____ __ _____
(И.О. Фамилия)
« ___ » _____ 202_)_ г.

ЗАДАНИЕ на курсовую работу

по дисциплине **РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ**

«ИНТЕРНЕТ-МАГАЗИН ЭЛЕКТРОННЫХ КНИГ»
(тема курсового проекта)

Студент _____ _____
(Фамилия, инициалы, номер группы)

График выполнения проекта: 25% к 4 нед., 50% к 8 нед., 75% к 11 нед., 100% к 14 нед.

1. Техническое задание

Разработанная РСОИ должна удовлетворять следующим требованиям:
использовать СОА (сервис-ориентированную архитектуру) для реализации системы;
посредством HTTP-запросов и/или очереди сообщений;
данные сервисов должны храниться в базе данных. Можно использовать SQL или NoSQL базы данных. Каждый сервис взаимодействует только со своей схемой данных;
выделить отдельный сервис и реализовать сбор статистики о пользовательских операциях;
реализовать пользовательский интерфейс и определить роли пользователей;
предусмотреть авторизацию пользователей, как через интерфейс приложения, так и через популярные социальные сети;
для запросов, выполняющих обновление данных на нескольких узлах распределенной системы, в случае недоступности одной из систем, выполнять полный откат транзакции;
реализовать хранение профиля пользователя в базе данных в хэшированном виде;
приложение должно поддерживать возможность горизонтального и вертикального масштабирования за счет увеличения количества функционирующих узлов совершенствования технологий реализации компонентов системы.

Оформление курсового проекта

2.1. Расчетно-пояснительная записка на 25-30 листах формата А4.

2.2. Перечень графического материала (плакаты, схемы, чертежи и т.п.)
_____

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель курсового проекта _____ _____

(Подпись, дата) (И.О.Фамилия)

Студент _____

_____

(Подпись, дата) (И.О.Фамилия)

Примечание:

Задание оформляется в двух экземплярах; один выдаётся студенту, второй хранится на кафедре.

Перечень типовых вопросов для защиты курсовой работы

Каким образом в случае недоступности части систем или внутренних ошибок сервера, возвращать ошибку с пояснением (т.е. http статус 4xx или 5xx + json с пояснением).

Как реализовать валидацию входных данных. Если данные имеют некорректный формат возвращать 400 Bad Request?

Каким образом для агрегирующего запроса на чтение в случае недоступности одной из систем, выполнять деградацию функциональности?

Как для одного из запросов, выполняющих обновление данных на нескольких системах, в случае недоступности одной из систем, выполнять полный откат операции?

Как для другого запроса, в случае недоступности одной из систем, возвращать пользователю успешный результат, а операцию ставить в очередь для повторного выполнения?

Как для всех операций, представленных через REST API, написать вызов с пользовательского интерфейса?

Как реализовать валидацию данных как на клиенте, так и на сервере?

Каким образом ошибки валидации отображать на форме и помечать некорректные поля с кратким пояснением?

Как в случае неуспешного ответа от сервера, выводить пользователю уведомление на странице или всплывающим окном?

Как все операции между сервером и клиентом разделить на публичные и закрытые. В REST API для выполнения закрытых операций использовать токен, полученный с помощью OAuth 2 (code flow)?

Каким образом пароль в базе хранить в хэшированном виде?

Как выделить отдельный сервис, отвечающий за авторизацию. Все запросы от UI и API на проверку токенов делегируютя ему через Aggregation Service?

Как выделить отдельный сервис, на который будут приходить информация с других сервисов через очередь (т.е. событийная модель). Среди этой информации должна быть оповещение об удачной/неудачной авторизации, запросах, изменяющих данные в системе и т.п.?

Каким образом реализовать три сводных отчета по полученным данным. Реализовать графическое представление этих отчетов?

Как добавить отдельную роль ADMIN, который будет иметь доступ к графическому представлению отчетов в специальной секции сайта, доступной только пользователям с этой ролью?

4.2. Процедуры оценивания знаний, умений, навыков, формы и организация текущего контроля и промежуточной аттестации обучающихся

Текущий контроль и промежуточная аттестации студентов в университете ведется

соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации студентов РГРТУ.

Текущий контроль успеваемости

Дисциплина в первом семестре делится на 3 модуля (включая Экзамен); во втором семестре состоит из 1 модуля(включая Курсовую работу). Каждый модуль учебной дисциплины включает в себя изучение законченного раздела, части дисциплины.

Основными видами контроля знаний, умений и навыков в течение каждого модуля учебной дисциплины являются рубежные контроли.

Текущий контроль по модулю учебной дисциплины осуществляется по графику учебного процесса. Сроки контрольных мероприятий (КМ) и сроки подведения итогов по модулям учебной дисциплины отображаются в рабочих учебных планах на семестр (отрезках). Студент должен выполнить все контрольные мероприятия, предусмотренные в модуле учебной дисциплины к указанному сроку, после чего преподаватель проставляет балльные оценки, набранные студентом по результатам текущего контроля модуля учебной дисциплины в ЭУ.

Контрольное мероприятие считается выполненным, если за него студент получил оценку в баллах, не ниже минимальной оценки, установленной программой дисциплины по данному мероприятию.

Студенты, не сдавшие контрольное мероприятие в установленный срок, продолжают работать над ним в соответствие с порядком, принятым кафедрой.

Промежуточная аттестация

Формой промежуточной аттестации в первом семестре является экзамен, во втором семестре является дифференцированный зачет.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Овечкин Геннадий Владимирович, Заведующий кафедрой ВПМ	**12.10.26** 09:03 (MSK)	Простая подпись
ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Овечкин Геннадий Владимирович, Заведующий кафедрой ВПМ	**12.10.26** 09:03 (MSK)	Простая подпись