

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский политехнический университет»

Инженерная школа информационных технологий и робототехники
Подразделение ОИТ
Направление, специальность 09.04.02 «Информационные системы и технологии»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка модуля интеграции системы трекинга задач YouTrack и корпоративной базы знаний на платформе XWiki

УДК 004.4:004.75:519.72

Студент

Группа	ФИО	Подпись	Дата
8ИМ71	Евстафьев Сергей Николаевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Евгений Александрович	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Копнов Максим Валерьевич	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Сосковец Любовь Ивановна	Д.И.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД	Атепаева Наталья Александровна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	К.Т.Н.		

**ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП ПРОГРАММЫ
МАГИСТРОВ 09.04.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»**

Код	Результат обучения (выпускник должен быть готов)
Общепрофессиональные и профессиональные компетенции	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
P5	Разрабатывать стратегии и цели проектирования, критерии

Код	Результат обучения (выпускник должен быть готов)
	эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
Р6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
Р7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
Р8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочным планировании.
Общекультурные компетенции	
Р9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
Р10	Свободно пользоваться русским и иностранным языками как средством делового общения.

Код	Результат обучения (выпускник должен быть готов)
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.02 «Информационные системы и технологии»
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

(Подпись) (Дата) Савельев А.О.
(Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ИМ71	Евстафьеву Сергею Николаевичу

Тема работы:

Разработка модуля интеграции системы трекинга задач YouTrack и корпоративной базы знаний на платформе XWiki

Утверждена приказом директора (дата, номер)	05.03.2019 г., № 1734/с
---	-------------------------

Срок сдачи студентом выполненной работы:	03.06.2019
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i>	Объектом исследования является система управлениями задач YouTrack, вики-система XWiki и способ их интеграции.
--	--

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Выявление цели и задач; Изучение предметной области; Разработка спецификации требований; Проектирование решения интеграции вики-системы XWiki и системы отслеживания задач YouTrack при помощи модуля расширения XWiki и веб-сервиса; Выбор технологий реализации; Реализация спроектированного решения; Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; Социальная ответственность.</p>
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ul style="list-style-type: none"> • Схема интеграции XWiki и YouTrack; • Детальная схема взаимодействия модуля интеграции и YouTrack; • IDEF-0 диаграмма процесса интеграции • Инфологическая модель (Уровень определений); • Инфологическая модель (Уровень ключей); • Инфологическая модель (Уровень атрибутов); • Функциональная диаграмма IDEF-0 • Физическая модель базы данных; • Алгоритм API метода создания задачи; • Алгоритм API метода обновления задачи; • Алгоритм API метода удаления задачи; • Страница авторизации административной панели; • Главная страница административной панели; • Презентация в формате *.pptx.
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Экономическая часть</p>	<p>Профессор ОСГН Сосковец Любовь Ивановна</p>

Социальная ответственность	Ст. преподаватель ООД Атепаева Наталья Александровна
Английская часть	Доцент ОИЯ Сидоренко Татьяна Валерьевна
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
2. Проектирование модуля интеграции (Designing integration module)	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	28.01.2019
---	------------

Задание выдал руководитель / консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Е.А.	к.т.н.		
Ст. преподаватель	Копнов М.В.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ71	Евстафьев С.Н.		

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский политехнический университет»

Инженерная школа информационных технологий и робототехники
Подразделение ОИТ
Уровень образования Магистратура
Направление, специальность 09.04.02 «Информационные системы и технологии»
Период выполнения Осенний/весенний семестр 2018/2019 учебного года

Формат представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполнения работы:	03.06.2019
---	------------

Дата контроля	Название раздела (модуля) виды работы (исследования)	Максимальный балл раздела (модуля)
10.03.2019	Теоретический анализ предметной области и постановка задач	20
15.04.2019	Проектирование модуля интеграции	30
21.05.2019	Реализация модуля интеграции	25
29.05.2019	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
16.05.2019	Социальная ответственность	5
29.05.2019	Обязательное приложение на иностранном языке	5

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Е.А.	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Копнов М.В.	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев А.О.	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИМ71	Евстафьеву Сергею Николаевичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	09.04.02 Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Оклад руководителя - 29568 руб. Оклад инженера - 21760 руб. Стоимость материальных ресурсов определяется на основе открытых прайсов.
2. <i>Нормы и нормативы расходования ресурсов</i>	–
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды 30% НДС 20 %

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого и инновационного потенциала НТИ</i>	Анализ конкурентных решений с позиции ресурсоэффективности и ресурсосбережения, оценка готовности к коммерциализации, диаграмма Исикавы, SWOT анализ.
2. <i>Разработка устава научно-технического проекта</i>	Цели и ожидаемые результаты исследования. Трудозатраты и основные функции исполнителей проекта.
3. <i>Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок</i>	Составление перечня этапов и работ, графика проведения работ, составление сметы затрат
4. <i>Определение ресурсной, финансовой, экономической эффективности</i>	Расчёт интегрального показателя финансовой эффективности и интегрального показателя ресурсоэффективности

Перечень графического материала (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> 1. Оценочная карта сравнения конкурентных интеграций 2. Бланк оценки степени готовности научного проекта к коммерциализации 3. Диаграмма Исикавы 4. Матрица SWOT-анализа проекта 5. Цели и результат проекта 6. График проведения работ 7. Смета затрат на разработку проекта 8. Риски проекта 9. Сравнительная оценка характеристик исполнения проекта

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Сосковец Любовь Ивановна	д.и.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ71	Евстафьев Сергей Николаевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа 8ИМ71	ФИО Евстафьеву Сергею Николаевичу
------------------------	---

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	09.04.02 Информационные системы и технологии

Тема ВКР:

Разработка модуля интеграции системы трекинга задач YouTrack и корпоративной базы знаний на платформе XWiki

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	<p>Объект исследования: модуль интеграции YouTrack.</p> <p>Область применения: использование модуля интеграции может осуществляться на вычислительных устройствах с локальной сетью.</p> <p>Цель исследования: интеграция вики-системы XWiki и YouTrack.</p>
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Правовые и организационные вопросы обеспечения безопасности:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> • Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 01.04.2019). • ГОСТ Р ИСО 6385-2016. Эргономика. Применение эргономических принципов при проектировании производственных систем. • ГОСТ 12.1.005-88 с изм. №1 от 2000 г., СанПиН 2.2.4.548-96. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
<p>2. Производственная безопасность:</p> <p>2.1. Анализ выявленных вредных и опасных факторов</p> <p>2.2. Обоснование мероприятий по снижению воздействия</p>	<ul style="list-style-type: none"> • Отклонение показателей микроклимата; • Превышение уровня шума; • Отсутствие или недостаток естественного света; • Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.
<p>3. Экологическая безопасность:</p>	<p>Влияние утилизации:</p> <ul style="list-style-type: none"> • Твердые отходы (системный блок компьютера, принтеры, сканеры, клавиатура, манипулятор «мышь») на литосферу; • Жидкие отходы (сточные воды) на гидросферу; • Люминесцентные лампы.
<p>4. Безопасность в чрезвычайных ситуациях:</p>	<p>Выбор типичной ЧС – пожар.</p> <ul style="list-style-type: none"> • разработка превентивных мер по предупреждению ЧС; • разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Атепаева Наталья Александровна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ71	Евстафьев Сергей Николаевич		

Реферат

Выпускная квалификационная работа состоит из 123 страниц, 23 рисунка, 32 таблицы, 26 источников, 7 приложений.

Ключевые слова: веб-сервис, проектирование, разработка, API, XWiki, YouTrack, интеграция.

Объектом исследования является разрабатываемая программная система для взаимодействия с системой трекинга задач YouTrack и корпоративной базой знаний на платформе XWiki.

Цель работы – создание модуля интеграции для взаимодействия системы трекинга задач YouTrack и корпоративной базы знаний на платформе XWiki.

В процессе исследования проводился анализ требований к модулю интеграции, сравнительный анализ платформ для разработки веб-приложений, анализ архитектур для реализации модуля интеграции, анализ существующих методов и проблем интеграции.

В результате исследования был спроектирован и разработан модуль интеграции для взаимодействия системы трекинга задач YouTrack и корпоративной базы знаний на платформе XWiki.

Степень внедрения: в данный момент модуль интеграции находится на этапе тестирования в ОАО «ТомскНИПИнефть».

Область применения: процесс разработки программного обеспечения в ОАО «ТомскНИПИнефть».

Экономическая эффективность/значимость работы: обусловлена важностью экономии времени в процессе разработки программного обеспечения.

Функциональность модуля интеграции может быть расширена добавлением дополнительных настроек для взаимодействия с другими системами управления задачами.

Определения, обозначения, сокращения, нормативные ссылки

ПП – программный продукт

JSON – JavaScript object notation

XML – eXtensible Markup Language

ИС – информационная система

API – Application Programming Interface

REST – Representational State Transfer

SOAP – Simple Object Access Protocol

RPC – Remote Procedure Call

Веб-сервис – это идентифицируемая веб-адресом программная система со стандартизированными интерфейсами

Модуль интеграции – это программное обеспечение, включающее в себя веб-сервис, административную панель и базу данных, и обеспечивающий взаимодействие системы трекинга задач YouTrack с вики-системой XWiki

Модуль расширения – это программное обеспечение, тесно интегрированное в платформу XWiki, позволяющее взаимодействовать с ней по её стандартам и API

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	17
1 ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ.....	19
1.1 Анализ литературы используемой в работе	19
1.2 Анализ проблем при интеграции информационных систем	20
1.3 Анализ существующих методов интеграции информационных систем... ..	24
1.3.1 Компоненты информационной системы	24
1.3.2 Интеграция платформ	26
1.3.3 Интеграция данных	27
1.3.4 Интеграция приложений	29
1.3.5 Интеграция бизнес-процессов	32
1.4 Цели работы и задачи	33
2 ПРОЕКТИРОВАНИЕ МОДУЛЯ ИНТЕГРАЦИИ	34
2.1 Структура системы трекинга задач YouTrack.....	34
2.2 Проектирование схемы взаимодействия систем.....	37
2.3 Построение функциональной диаграммы с декомпозицией	39
2.4 Проектирование реляционной модели базы данных.....	42
2.5 Разработка спецификации требований к системе.....	44
2.5.1 Требования к системе	44
2.5.2 [F] Общие функциональные требования	45
2.5.3 [P] Требования к средствам интеграции.....	46
2.5.4 [IS] Требования к информационной безопасности.....	47
2.5.5 [TS] Требования к техническому обеспечению	48
2.5.6 [SR] Требования к программному обеспечению	48
2.5.7 [RG] Требования к надежности	49
2.5.8 [D] Требования к документации.....	49
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДУЛЯ ИНТЕГРАЦИИ.....	51
3.1 Обзор шаблонов проектирования архитектур системы	51
3.2 Архитектурный подход для организации веб-сервиса.....	54

3.3	Выбор языка программирования и среды разработки	55
3.4	Выбор инструментов для разработки	57
3.5	API системы YouTrack.....	61
3.6	Описание разработанных методов системы.....	64
3.7	Особенности разработанного веб-сервиса	65
4	ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТИ И РЕСУРСОСБЕРЕЖЕНИЕ.....	69
4.1	Предпроектный анализ	69
4.1.1	Анализ конкурентных технических решений	69
4.1.2	Оценка готовности проекта к коммерциализации.....	71
4.1.3	Диаграмма Исикавы.....	73
4.1.4	SWOT-анализ.....	74
4.2	Инициация проекта	77
4.2.1	Устав проекта	77
4.2.2	Ограничения и допущения проекта	77
4.2.3	Организация и планирование работ.....	78
4.2.4	Продолжительность этапов работ	78
4.3	Расчет сметы затрат на выполнение проекта	80
4.3.1	Расчет материальных затрат	81
4.3.2	Расчет заработной платы	81
4.3.3	Расчет затрат на электроэнергию	82
4.3.4	Расчет затрат на социальные нужды	83
4.3.5	Расчет амортизационных расходов	83
4.3.6	Расчет накладных расходов	84
4.3.7	Расчет планируемой прибыли.....	84
4.3.8	Расчет НДС	84
4.3.9	Цена разработки	85
4.3.10	Расчет общей себестоимости разработки	85
4.4	Потенциальные риски.....	85
4.5	Определение ресурсной и финансовой эффективности проекта	86

4.6 Выводы по разделу.....	88
5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	89
5.1 Правовые и организационные вопросы обеспечения безопасности	89
5.2 Производственная безопасность	92
5.2.1 Отклонение показателей микроклимата.....	93
5.2.2 Превышение уровня шума	94
5.2.3 Отсутствие или недостаток естественного света.....	95
5.2.4 Электромагнитное излучение	97
5.3 Экологическая безопасность.....	97
5.4 Безопасность в чрезвычайных ситуациях.....	98
ЗАКЛЮЧЕНИЕ	100
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	101
ПРИЛОЖЕНИЕ А	104
ПРИЛОЖЕНИЕ Б.....	118
ПРИЛОЖЕНИЕ В	119
ПРИЛОЖЕНИЕ Г.....	120
ПРИЛОЖЕНИЕ Д.....	121
ПРИЛОЖЕНИЕ Е.....	122
ПРИЛОЖЕНИЕ Ж	123

ВВЕДЕНИЕ

Вопросы интеграции приложений предприятия активно обсуждаются сегодня компьютерным сообществом. Не существует информационных систем, которые в одиночку могли бы покрыть потребности современного предприятия. Средние и крупные организации обычно эксплуатируют как минимум десяток многопользовательских систем, а иногда счет идет на сотни и тысячи. В этих системах часто обрабатываются одинаковые данные — начиная со справочников и классификаторов, от чего задача интеграции на сегодняшний день является одной из важнейших [1].

Интеграция информационных систем — это процесс установки связей между информационными системами предприятий для получения единого информационного пространства и организации поддержки бизнес-процессов.

В настоящее время применяется значительное число программ для управления задачами в IT-разработке. Они могут использоваться по-разному в каждой отдельной компании. Часть из них могут использоваться лишь для баг-трекинга, но большинство используется только в качестве отслеживания задач [2].

Предприятие ОАО «ТомскНИПИнефть» – это ведущий корпоративный научно-проектный институт в области комплексного проектирования для предприятий нефтегазового комплекса. В институте ведется разработка профильного программного обеспечения для предприятий нефтегазового комплекса, где в ходе работы используются различные инструменты. В частности спецификация требований на предприятии ведётся в XWiki, а трекинг задач осуществляется с помощью YouTrack.

Поскольку на текущий момент в ОАО «ТомскНИПИнефть» вики-система XWiki и YouTrack не имеют интеграции друг с другом, то существует проблема переноса этих требований из одной системы в другую с минимальными временными затратами. Эта проблема является актуальной на сегодняшний день и требует автоматизации.

Данная работа является частью общего проекта по интеграции систем XWiki и YouTrack, отражающая взаимодействие между системой трекинга задач YouTrack и модулем интеграции. В работе будут использоваться материалы, представленные при проектировании взаимодействия между XWiki и модулем интеграции, который является общим посредником при интеграции систем проекта.

Целью данной работы является обеспечение интеграции таких бизнес-процессов предприятия, как разработка программного обеспечения и выявление требований к разработке ПО, путем создания модуля интеграции, предназначенного для передачи и обработки данных от вики-системы XWiki в систему трекинга задач YouTrack посредством своего API.

В разделе 1 проведен анализ литературы, используемой в работе, проанализированы проблемы и существующие методы интеграции информационных систем.

Раздел 2 содержит описание этапа проектирования модуля интеграции. Определена структура системы трекинга задач YouTrack, построены IDEF0 диаграммы процесса интеграции, общая схема взаимодействия двух систем и детальная схема взаимодействия разработанного веб-сервиса, в рамках реализации модуля интеграции, и системы YouTrack. Спроектирована реляционная модель базы данных веб-сервиса. Разработана спецификация требований к веб-сервису.

В 3 разделе описывается программная реализация всех составляющих компонентов модуля интеграции. Обоснован выбор шаблона проектирования архитектур системы, языка программирования и инструментов для разработки. Описаны разработанные методы веб-сервиса и представлены его особенности.

Раздел 4 посвящен вопросам финансового менеджмента, ресурсоэффективности и ресурсосбережения. Был произведен расчет затрат на разработку и вычислена экономическая эффективность проекта.

В 5 разделе содержится анализ действующих стандартов безопасности труда при разработке и эксплуатации веб-сервиса.

1 ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

Перед тем как приступить к проектированию и разработке программного продукта, необходимо провести обзор предметной области.

В данной главе будет проведён анализ теоретических и практических работ по теме интеграции информационных систем. Показана актуальность решаемой задачи, проведен анализ используемой литературы, также поставлена цель данной работы и определены основные задачи.

1.1 Анализ литературы используемой в работе

Литературу по основной части работы ВКР можно условно разделить на три группы.

В первую группу можно включить источники, содержащие различную информацию для проведения анализа предметной области [1, 2, 3, 5, 6, 11, 13, 14].

Во вторую группу следует включить источники, описывающие методологии решений [4, 7, 8, 9, 16]. Данная группа является наиболее важной для магистерской выпускной квалификационной работы, так как на основе этих методологий можно создать модуль интеграции, отвечающий всем требованиям разработки. Статьи и публикации из этой группы были взяты из известных и популярных ресурсов, посвящённых разработке программных продуктов и их сопровождения.

К третьей группе источников относятся статьи, содержащие конкретные примеры интеграционных решений [условно 9, 12, 15].

Проведённый анализ показал, что выделенные группы наиболее полно раскрывают предметную область и сущность проблемы. Первая группа литературы все специфические термины, характеризующие выбранные технологии. Вторая группа описывает общие понятия и стандарты разработки

веб-сервиса. А третья дополнит теоретическую часть работы фактами, примерами и актуальными проблемами, связанными с разработкой.

1.2 Анализ проблем при интеграции информационных систем

На сегодняшний момент интеграция информации имеет большое значение для большинства крупных предприятий по всему миру. Под ней воспринимается интеграция данных, которые могут находиться в огромном множестве систем одновременно и предоставляться в стандартизированном виде. Такая информация необходима для изучения, анализа и обработки данных. В целом, любая подобная интеграция рассчитана только для конечных пользователей системы, так как им придется в будущем работать с этим множеством различных систем, решая различные задачи.

Тут необходимо отметить, что важным условием реализации любой интеграции информации значится тщательный анализ для систем, с которыми будет работать конечный пользователь, а также самих данных. Всё это необходимо для понимания того, являются ли эти данные релевантные? Так как если это не так, то и незачем их извлекать из источников и как-либо преобразовывать. Кроме того, анализ должен проводиться для целевых структур, куда данные будут заноситься в будущем после обработки и соответствующих преобразований. В последующем, применяя аналитические средства будет составляться отчет, на основании которого конечный пользователь сможет принимать решения по актуальным задачам.

В системах данные могут вноситься и иметь вид совершенно различный, нежели в аналогичных системах от других предприятий. Чтобы извлечь какую-либо выгоду от таких данных, необходимо их сопоставлять между собой. Эту задачу обычно решает конечный пользователь, от чего могут возникнуть повторяющиеся, либо вовсе неверные данные. При их использовании системами может возникнуть множество проблем и несоответствий.

При интеграции множества источников данных многократно возрастает нужна в очистке или преобразований данных, например, при помощи

алгоритма Левенштейна. Это одно из основных решений проблемы качества предоставляемой нам информации, хранящейся в базе данных. Ведь, искаженные данные, содержащие в себе различные опечатки при вводе или неверно форматированные строки, могут существенно усложнить процесс анализа данных конечным пользователем, так как различные источники часто содержат в себе данные, имеющие разный формат или вид. Для качественного анализа поступивших данных важно использовать консолидацию данных и исключать дубликаты.

В настоящее время базы данных составляют огромный процент от всех видов хранилищ, используемые под хранение различных данных, с которыми ежедневно проводятся разные операции или загружаются новые данные. Базы данных идеально подходят для работы с большими объемами данных из разных систем. Однако, с этим возрастают риски по получению некорректных данных в базу при обработке данных из множества различных источников. Необходимо постоянно корректировать поступившие данные во избежание дубликатов или не отформатированной информации, которая может существенно повлиять на общую статистику системы, особенно если она важна для конечного пользователя и используется для принятия каких-либо решений. Но, очистка данных это не просто средство устранения всех проблем, это одна из самых больших проблем в интеграции. Она выполняется лишь в каждой конкретной выборке, которую подготавливают разработчики и этот процесс является достаточно трудоемким. Конечному пользователю поступают уже готовые для работы данные.

Любые изменения в структуре базы данных требуют преобразование самих данных. Они становятся особенно необходимы, когда осуществляется переход на другую платформу, либо добавляются новые источники данных в множество, с которым должна работать система. Важно понимать, что изменение представления или содержания данных также потребует произвести трансформацию данных в более подходящий формат.

В случае интеграции множества систем, следует понимать, что проблема в отдельной системе может вылиться в общую проблему для всех интегрированных систем. Ни одна система не застрахована от некорректных данных, которые могут туда попасть при внедрении системы на предприятия, у которых могут иметься специфические потребности, или поддержки других систем. В итоге появляется серьезная неоднородность в текущих данных, моделях и системах управления данными. Такая неоднородность затрудняет работу с данными и делает невозможным проведение анализа и принятия на основе него какого-либо решения конечному пользователю.

Информационные системы, на сегодняшний момент, обладают некоторыми подсистемами, разрабатываемыми для очистки данных. Однако, стоит смотреть сразу в корень проблемы и не допускать ввода некорректных данных в систему. Такими средствами, ограничивающими пользовательский ввод, обладают многие системы и в том числе различные СУБД. Следует позаботиться и о том, каким образом будет проводиться очистка данных. Вероятно, можно установить правила, которым будет следовать алгоритм очистки для улучшения ограничений имеющихся схем. На этапе проектирования необходимо тщательно проработать схему базы данные, ограничение целостности и приложение-клиент, через которое будет возможность вносить данные в систему.

Выделяют следующие основные конфликты, с которыми сталкиваются разработчики на этапе проектирования базы данных:

- Конфликты наименований (конфликт возникает, когда одно и то же имя используется для различных отношений, а также когда используются различные имена для одного отношения);
- Конфликты структур или структурные конфликты (конфликт возникает, когда один и тот же объект имеет разные представления в разных источниках).

Эти конфликты встречаются на уровне схемы базы данных, однако огромное множество конфликтов может встречаться исключительно на уровне

элемента данных. К ним относятся дубликаты, некорректные данные и т.д. Данные или представления одного и того же имени атрибута могут также различаться из разных источников. Кроме того, данные могут быть по-разному агрегированы или представлены за различные временные промежутки.

Поиск «перекрывающихся» данных, безусловно, является главной задачей при интегрировании данных из нескольких систем. В частности, перекрывающимися данными называют идентичные данные из систем, которые принадлежат одному и тому же объекту в различных источниках, как показано на рисунке 1.1.

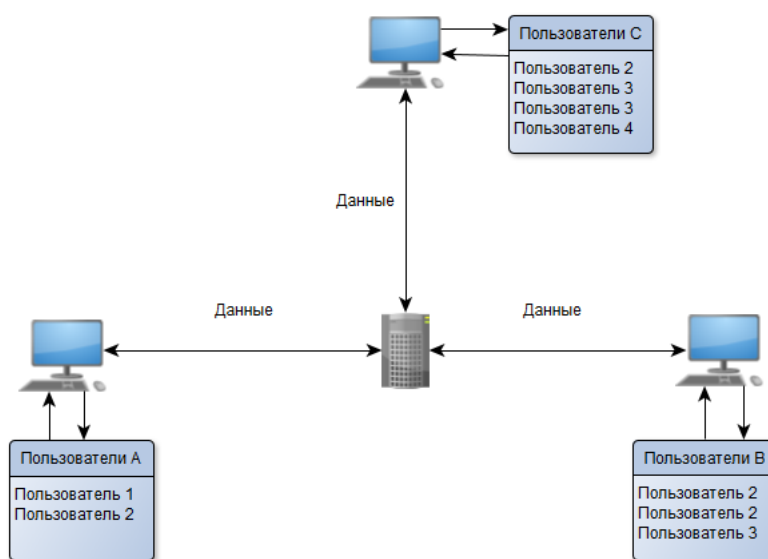


Рисунок 1.1 – Дублирование данных

Эта проблема в кругах специалистов, на сегодняшний день, именуется как проблема идентичности объекта. Она опасна еще и тем, что смешивает релевантные данные с некорректными или устаревшими идентичными данными. Неправильная очистка такого массива данных может повлечь за собой потери некоторых частей данных. Данные в системе лишь местами избыточны и могут дополняться другими данными из других систем, получая наиболее полную информацию об определенном объекте реального окружения. Дополняющие данные обязательно должны консолидироваться для правильного представления информации конечному пользователю.

Для того чтобы успешно решать все интеграционные проблемы, описанные в данном разделе, требуется правильно интегрировать данные на уровне схемы и элементов одновременно и корректно выполнять очистку данных. Проблемы необходимо решать в строгой очередности. Для начала стоит начинать решать структурные конфликты на уровне схемы. Затем плавно переходить к более локальным проблемам на уровне элементов данных, решая проблему идентичности данных и избавляясь от дубликатов. В конце концов, правильно структурированная и согласованная информация из различных источников это уже решение половины всех проблем [3].

1.3 Анализ существующих методов интеграции информационных систем

1.3.1 Компоненты информационной системы

Информационная система представляет собой совокупность нескольких компонентов, поэтому, говоря об интеграции информационных систем, следует подразумевать интеграцию составляющих их компонентов.

Обычно, информационная система содержит в себе следующие компоненты, представленные на рисунке 1.2:

- Платформа, на которой функционируют остальные компоненты системы, включающая в себя аппаратуру (железо) и системное ПО;
- Данные, с которыми работает система. Состоят из СУБД и баз данных;
- Приложения, реализующие бизнес-логику по работе с данными системой. Состоят из компонентов бизнес-логики, пользовательского интерфейса, вспомогательных компонентов (фреймворк) и сервера приложений, который обеспечивает хранение и доступ к компонентам приложения;
- Бизнес-процессы, представляющие из себя сценарии работы пользователей с системой.

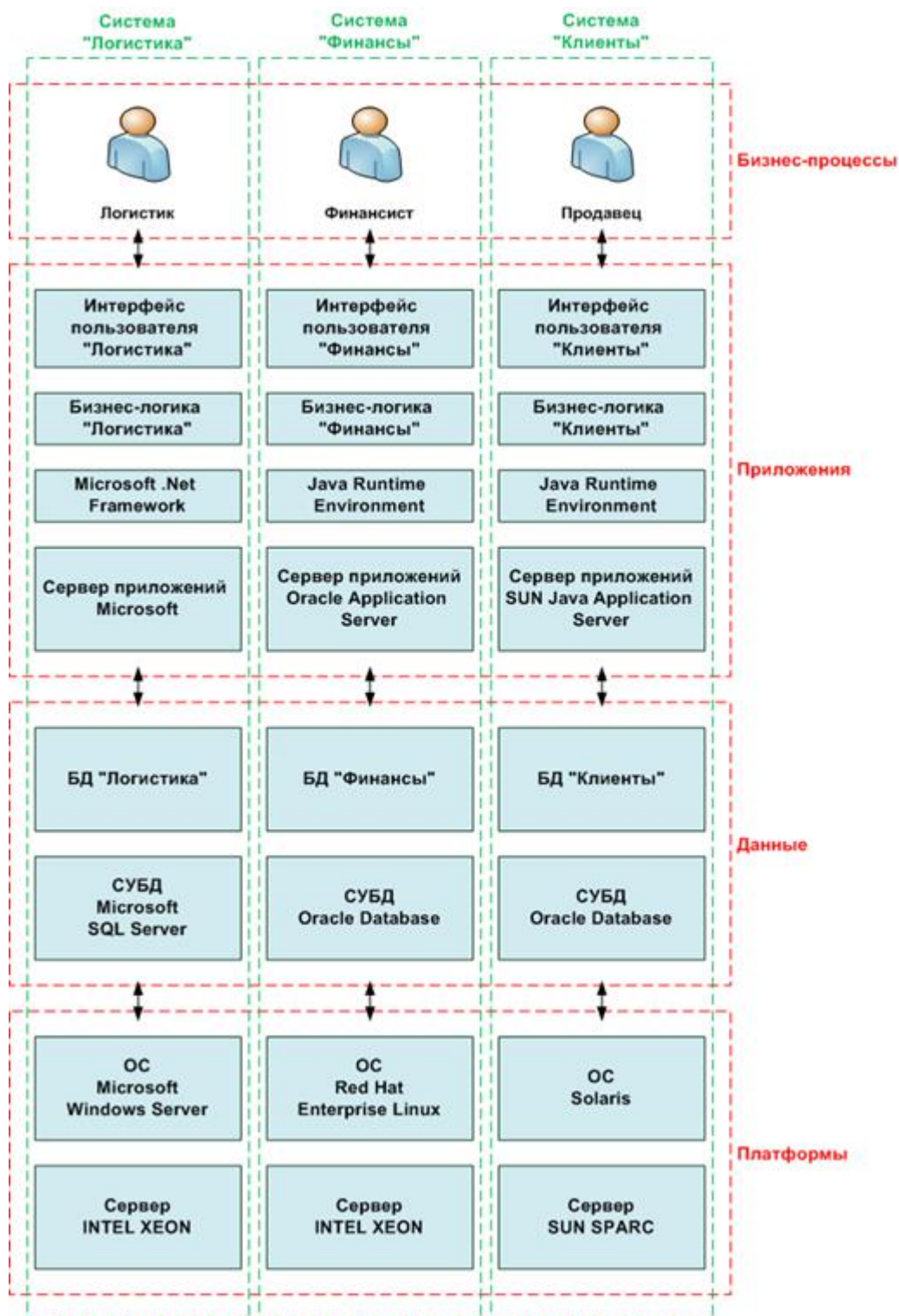


Рисунок 1.2 – Основные компоненты информационных систем

Поэтому, интеграция информационных систем заключается в интеграции одного или нескольких компонентов интегрируемых информационных систем (объектов интеграции).

1.3.2 Интеграция платформ

Целями интеграции платформ являются:

- Обеспечение возможности взаимодействия между приложениями, работающими на различных программно-аппаратных платформах (например, между приложениями, работающими на серверах Windows, Solaris, Linux и др.);
- Обеспечение возможности работы приложений, разработанных для одной программно-аппаратной платформы, на других программно-аппаратных платформах (например, приложений Windows на платформах Linux, Solaris и др.).

Существует несколько подходов, направленных на достижение этих целей. В рамках каждого из подходов существуют различные технологии:

- Удаленный вызов процедур (RPC, Web-сервисы, REST и пр);
- ПО промежуточного слоя (Microsoft.Net, Java Runtime);
- Виртуализация.

Технологии удаленного вызова процедур (в широком смысле под процедурой понимается некоторая функциональность приложения) позволяют опубликовать процедуру и обеспечить возможность ее вызова (передачи входящих параметров и получения выходных результатов) для приложений, работающих на других платформах.

Концепция программного обеспечения промежуточного слоя (framework, среда исполнения, виртуальная машина) состоит в разработке прикладного ПО не с использованием сервисов конкретной операционной системы (например, Windows API), а с использованием сервисов ПО промежуточного слоя. Разработчиками ПО промежуточного слоя создаются ее реализации под различные операционные системы, которые транслируют вызовы соответствующих функций фреймворка в вызовы соответствующей операционной системы. Типичным примером является технология Java Runtime Environment. Приложения, разработанные для этой технологии работают на

любых программно-аппаратных платформах (Windows, Linux и др.) без каких-либо доработок самих приложений. Аналогичные возможности предоставляет среда Microsoft .Net Framework.

Интересной и современной концепцией является «виртуализация». К интеграции платформ она имеет отношение постольку, поскольку позволяет существенно упростить использования различных платформ и, соответственно, использование систем, требующих для своего функционирования наличия конкретных платформ. Если без виртуализации возможно одновременное функционирование N операционных сред на N серверов, то применение технологий виртуализации позволяет обеспечить функционирование N операционных сред на M серверов. Если $N > M$ – это позволяет сократить расходы на аппаратное обеспечение путем его более эффективного использования. Если $N < M$ – это простой путь увеличения производительности систем. Например, виртуализация позволяет развернуть и одновременно использовать на одном физическом сервере несколько операционных систем: Windows, Linux и др. На каждом из таких «виртуальных» серверов могут быть развернуты соответствующие системы, которые будут доступны одновременно. Примеры технологий виртуализации: Microsoft Hyper-V, KVM, OpenVZ, Virtuozzo, VMware, Xen и пр.

1.3.3 Интеграция данных

По определению информационная система работает с данными. В подавляющем большинстве случаев система имеет в своем составе базу данных для их хранения. Интеграция на уровне данных предполагает совместное использования данных различных систем. Интеграция данных может оказаться проще, чем интеграция приложений, которая будет описана далее, т.к. промышленные СУБД, в которых обычно хранят данные информационные системы, имеют развитые возможности программного доступа к данным из других приложений. Сами приложения при этом могут иметь весьма ограниченные возможности программного (вне собственного

пользовательского интерфейса) использования своей функциональности внешними системами.

Подходы к интеграции данных:

- Универсальный доступ к данным;
- Хранилища данных.

Технологии универсального доступа к данным позволяют обеспечить единообразный доступ к данным различных СУБД. Посредником для работы с конкретной СУБД в данном случае является драйвер для соответствующей СУБД. Например, один и тот же SQL-запрос на выборку данных `SELECT * FROM TTABLE` может быть использован на выборку данных из таблицы `TTABLE`, хранящейся в различных СУБД. Это позволяет абстрагироваться от специфики конкретных СУБД и легко осуществлять интеграцию данных, хранящихся в различных СУБД. Наиболее распространенные технологии этого класса: ODBC, JDBC, ADO.NET. Кроме того, на сегодняшний день широко распространены технологии объектно-реляционного отображения (ORM), которые также позволяют абстрагироваться от деталей взаимодействия с конкретными СУБД. Примерами таких технологий являются Entity Framework, Hibernate, NHibernate, Flexberry ORM.

Концепция хранилищ данных состоит в создании корпоративного хранилища данных. Хранилище данных – база данных, хранящая в себе данные, собираемые из баз данных различных информационных систем, для целей их дальнейшего анализа. Например, может быть создано единое хранилище данных компании, в которое собрана информация из бухгалтерской, оперативной системы, внешних систем партнеров компаний. Для создания хранилищ данных используются технологии (OLAP), отличные от технологий создания оперативных БД (OLTP). В основном это делается для повышения производительности выполнения сложных аналитических запросов по многим параметрам (многомерные запросы). Подходы к созданию и наполнению хранилищ данных отражены в парадигме ETL (extraction, transformation, loading = извлечение, преобразование и загрузка). Технологии и инструментальные

средства анализа больших массивов данных с целью выявления закономерностей предметной области объединяются понятием «Data Mining». Термин для совокупности технологий хранилищ данных и инструментальных средств, обеспечивающих перевод транзакционной деловой информации в человекочитаемую форму, пригодную для бизнес-анализа – «Business Intelligence».

1.3.4 Интеграция приложений

Интеграция на уровне приложений подразумевает использование готовых функций приложений другими приложениями. Например, разрабатывая систему электронного документооборота, существует возможность использовать в рамках этой системы в качестве текстового редактора MS Word вместо того, чтобы разрабатывать свой собственный текстовый редактор. Или, например, ПО Call-центра, получив входящий звонок от клиента, имеет возможность обратиться к функции биллинговой системы по проверке баланса (на входе – номер телефона абонента, на выходе – его текущий баланс) и, в зависимости от состояния баланса соединить его с оператором или автоматически проинформировать о необходимости пополнить свой счет. При этом структура база данных биллинговой системы является ее внутренней информацией, публикуются конкретные функции, позволяющие другим системам работать с конкретными данными.

Стоит упомянуть следующие подходы к интеграции приложений:

- Интерфейсы прикладного программирования;
- Обмен сообщениями (Корпоративная сервисная шина);
- Сервис-ориентированная архитектура;
- Интеграция пользовательских интерфейсов.

Программный интерфейс конкретной системы представляет собой «опубликованный» функционал этой системы, который может быть использован извне. Функционал может публиковаться в виде набора функций

(пример – Windows API) или в виде объектной модели (объекты со свойствами и методами, пример – объектные модели приложений Microsoft Office).

В большинстве случаев интеграция нескольких систем заключается в передаче информации между ними, например, в форме запрос-ответ. Если системы функционируют в гетерогенных распределенных средах, то принципиальное значение имеет обеспечение гарантированности, безопасности, управляемости доставки информации между приложениями. Эти и другие принципы реализуются в корпоративных системах обмена сообщениями. В данном случае речь идет об обмене сообщениями между приложениями, а не людьми, как, например, в случае E-mail или мессенджеров. Функциональность этих систем достаточно прозрачна – прием сообщения от одного приложения, транспортировка по заданным правилам и передача этого сообщения другому приложению. При этом может производиться шифрование сообщений (для невозможности прочтения данных в процессе транспортировки), цифровая подпись (для защиты от умышленного изменения данных во время пути сообщения), настройка подписки (для отправки одного сообщения сразу нескольким приложениям), определение метаданных для сообщений (для облегчения использования сообщений со сложной структурой содержимого) и др.

Сервис-ориентированная архитектура (SOA) является логическим продолжением концепции веб-сервисов, которая состоит в публикации функциональных блоков какого-либо приложения в виде, позволяющем получить к ним доступ другим приложениям через Веб. Веб (протокол HTTP) в данном случае привлекателен ввиду возможности его использования и, соответственно, использования опубликованного в Веб-приложений функционала на любых программно-аппаратных платформах. Веб-сервис – небольшая программная надстройка над функционалом приложения, преобразующая вызовы, получаемые через Веб, во внутренние вызовы функций приложения и возвращающая результаты обратно. Основными идеями SOA являются:

- Публикация функционала корпоративных приложений в виде Веб-сервисов. Упорядочивание опубликованных сервисов в виде каталога;
- Построение на основе Веб-сервисов новых приложений путем их комбинации.

Цена создания новых приложений на основе существующих Веб-сервисов будет существенно ниже, чем разработка приложений «с нуля» или обширная интеграция с другими системами.

Например, в компании (оператор связи) существует система Service Desk (техническая поддержка абонентов) и биллинговая система (тарификация услуг). Перед компанией стоит задача сделать новую систему «Личный кабинет абонента», в которой абонент мог бы через Интернет просмотреть состояние своего счета и сообщить о неисправности. Для этого компания вместо того, чтобы создавать «Личный кабинет» с собственной базой данных, синхронизируемой с БД биллинговой системы и системы Service Desk, использует готовые Web-сервисы «Карточка абонента» (опубликованный функционал биллинговой системы) и «Создать заявку в техподдержку» (опубликованный функционал системы Service Desk). Очевидно, что вся работа по новому приложению «Личный кабинет» состоит лишь в создании Веб-интерфейса пользователя на сайте компании.

Также часто используется следующий подход – интеграция пользовательских интерфейсов. Например, для создания приложений «одного окна». Простейший пример – фреймы на Веб-странице. Внутри каждого фрейма содержится отдельное Веб-приложение. Благодаря фреймам, все эти приложения отображаются на экране одновременно. Пользовательские интерфейсы Веб-приложений легко интегрируются, однако, существуют возможности интегрировать и «классические» пользовательские интерфейсы и их фрагменты (ActiveX).

1.3.5 Интеграция бизнес-процессов

Наиболее целостным подходом к интеграции систем является интеграция на уровне бизнес-процессов. В рамках интеграции бизнес-процессов происходит и интеграция приложений, и интеграция данных и, что не менее важно, людей, вовлеченных в этот бизнес-процесс. Интеграция на уровне бизнес-процессов является наиболее «естественной» для организаций, так как их деятельность состоит, прежде всего, именно из бизнес-процессов, а не приложений, баз данных и платформ.

Идеи, лежащие в основе интеграции бизнес-процессов, достаточно просты:

- Составить сценарий некоторого бизнес-процесса, происходящего в организации, описать в нем операции взаимодействия пользователей с различными системами и систем между собой. Таким образом, бизнес-процесс является элементом, логически интегрирующим различные системы. Сценарий создается при помощи специализированного программного продукта, который далее будет управлять ходом этого бизнес-процесса согласно сценарию.

- Операции взаимодействия с системами в рамках бизнес-процесса детально описываются в терминах информационного обмена: форматы обмена, используемые сервисы, приложения, события, правила, политики и т.п.

- К интегрирующему программному обеспечению, при помощи которого описан сценарий бизнес-процесса, подключаются посредством адаптеров интегрируемые системы, вовлеченные в бизнес-процесс. Таким образом, становится возможным автоматизированный информационный обмен между системами.

- Готовый к выполнению бизнес-процесс выводится на «пульт управления» менеджера, при помощи которого, он может запускать и останавливать бизнес-процессы, отслеживать их состояние, вводить данные и принимать решения на отдельных операциях бизнес-процессов, требующих

участия человека и др. Взаимодействия между системами, не требующее участия человека осуществляется автоматически интегрирующим ПО [4].

1.4 Цели работы и задачи

Целью данной магистерской выпускной квалификационной работы является разработка модуля интеграции, предназначенного для передачи и обработки данных от вики-системы XWiki в YouTrack по средствам своего API.

В модуль интеграции входит веб-сервис, административная панель и база данных для подсистемы журналирования.

Для достижения поставленной цели, требуется решить следующие задачи:

- Выявление требований для разработки модуля интеграции;
- Проектирование модуля интеграции для взаимодействия вики-системы XWiki и системы трекинга задач YouTrack;
- Разработка спроектированного модуля интеграции.

2 ПРОЕКТИРОВАНИЕ МОДУЛЯ ИНТЕГРАЦИИ

После того как проведен анализ и обоснована необходимость выполнения данной работы, необходимо произвести проектирование модуля интеграции для его последующей реализации. Для этого будет рассмотрена система трекинга задач YouTrack и её структура для описания возможностей взаимодействия с модулем интеграции.

2.1 Структура системы трекинга задач YouTrack

YouTrack – это коммерческая система отслеживания ошибок, программное обеспечение для управления задачами и проектами, разработанное компанией JetBrains. YouTrack поддерживает поисковые запросы, автодополнение, манипуляцию с наборами задач, настройку набора атрибутов задачи, создание пользовательских рабочих процессов и реализует подход, основанный на преимущественном использовании клавиатуры. Интерфейс программы представлен на рисунке 2.1.

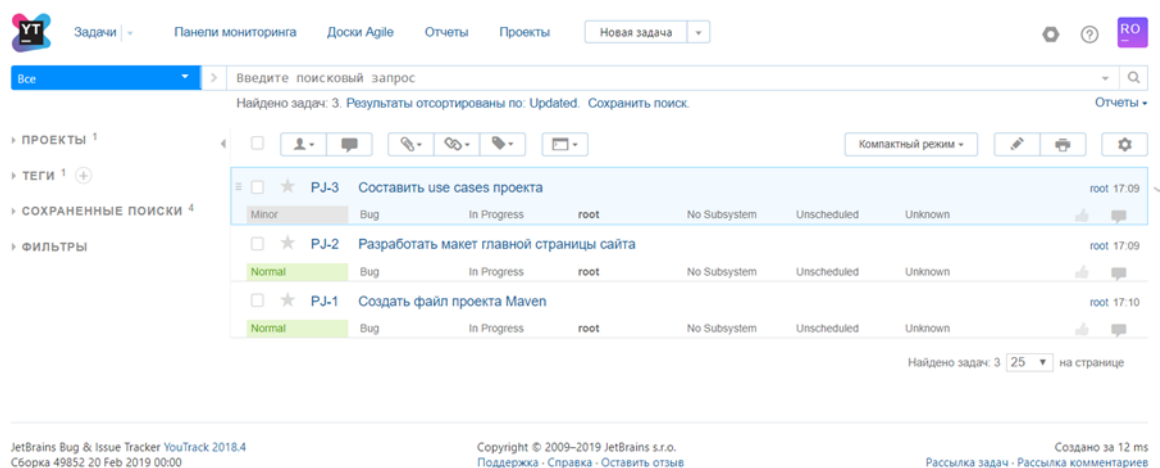


Рисунок 2.1 – Интерфейс системы

Система YouTrack разработана по парадигме ЯОП (языково-ориентированное программирование). Система использует свои собственные предметно-ориентированные языки программирования, разработанные компанией JetBrains, а также JetBrains MPS. Использует собственную базу

данных под названием JetBrains Database, где данные представляются парами в виде ключа и значения. Через собственный веб-сервис YouTrack позволяет производить удаленные вызовы процедур через REST API.

YouTrack имеет встроенную поддержку интеграции с множеством систем управления версиями, включая Git, CVS и другие. Благодаря git есть возможность интеграции и с сервисом GitHub. Система также имеет интеграцию на уровне приложений с такими системами как TeamCity, IntelliJ IDEA и т.д. Поддерживает аутентификацию пользователей через аккаунты Google и Yahoo!. Имеется возможность импортировать задачи из других систем трекинга, таких как JIRA. На этот случай у YouTrack существует целая библиотека по импорту задач, которой может воспользоваться любой желающий. Она написана на Python [5].

В версии YouTrack inCloud процесс настройки системы очень простой. Необходимо лишь зарегистрироваться на официальном сайте и иметь постоянное интернет соединение. Все данные будут храниться в облаке с ежедневным резервным копированием данных.

YouTrack распространяется под коммерческой лицензией в пяти различных вариантах, различающихся количеством поддерживаемых пользователей, которые могут работать с системой. JetBrains предоставляет YouTrack для бесплатного использования разработчикам открытых проектов и для обучения [6].

Веб-сервис работает с двумя стандартными сущностями, предусмотренными структурой YouTrack: задачами и проектами. В таблице 2.1 представлены атрибуты сущности Project.

Таблица 2.1 – Атрибуты сущности Project

Атрибут	Тип	Описание
startingNumber	Числовой	Стартовый номер ID, при создании новых задач
shortName	Строковый	ID проекта
description	Строковый	Описание проекта

Продолжение таблицы 2.1

leader	Объект User	Руководитель проекта
createdBy	Объект User	Создатель проекта
issues	Массив	Список задач проекта
customFields	Массив	Список пользовательских атрибутов
archived	Логический	Нахождение проекта в архиве
template	Логический	Проект является шаблоном
iconUrl	Строковый	Путь к иконке проекта

В таблице 2.2 представлены атрибуты сущности Issue.

Таблица 2.2 – Атрибуты сущности Issue

Атрибут	Тип	Описание
idReadable	Строковый	ID задачи
created	Числовой	Дата создания задачи
updated	Числовой	Дата последнего обновления задачи
resolved	Числовой	Дата решения задачи
numberInProject	Числовой	Порядковый номер задачи
project	Объект Project	Проект, в рамках которого создана задача
summary	Строковый	Заголовок задачи
description	Строковый	Описание задачи
reporter	Объект User	Создатель задачи
updater	Объект User	Последний пользователь, обновивший задачу
visibility	Объект Visibiliy	Настройки видимости задачи различным группам пользователей
votes	Числовой	Количество голосов за задачу
comments	Массив	Список комментариев к задаче
commentsCount	Числовой	Количество комментариев
tags	Массив	Список тегов к задаче

Продолжение таблицы 2.2

links	Массив	Список ссылок, ассоциированных с задачей
customFields	Массив	Список пользовательских атрибутов
voters	Объект IssueVoters	Информация о голосовавших пользователей
watchers	Объект IssueWatchers	Информация о пользователях, отслеживающих задачу
attachments	Массив	Список материалов к задаче
subtasks	Объект IssueLink	Список подзадач данной задачи
parent	Объект IssueLink	Родительская задача данной задачи

В системе YouTrack имеется более 70 различных сущностей, представляющие собой объекты со своими атрибутами. REST API YouTrack позволяет работать с большинством из них, однако для решения задач в последующем будем рассматривать лишь работу с сущностями Project и Issue, так как они являются общими объектами при интеграции двух систем.

2.2 Проектирование схемы взаимодействия систем

Подключение модуля XWiki происходит посредством API XWiki и менеджера расширений. Взаимодействие расширения XWiki с модулем интеграции осуществляется посредством API расширения и API веб-сервиса. Данные передаются расширением XWiki в формате JSON на веб-сервис, где они анализируются и передаются на YouTrack посредством REST API.

Общая схема интеграции представлена на рисунке 2.2.

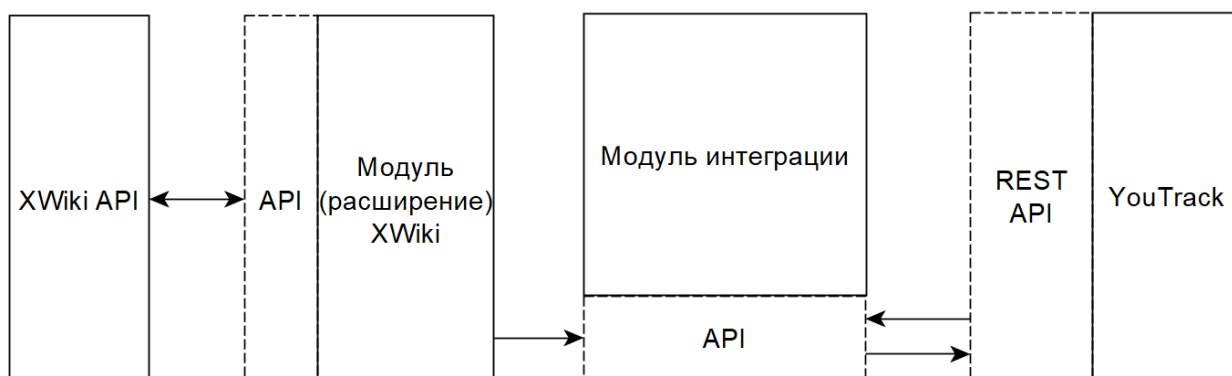


Рисунок 2.2 – Схема интеграции XWiki и YouTrack

Перед отправлением данных в YouTrack при создании новой задачи веб-сервис добавляет в таблицу соотношений ID задачи YouTrack и название задачи XWiki, от которой они поступили.

При обновлении данных, на веб-сервисе анализируются приходящие данные адреса страницы XWiki, сравниваются по таблице соотношений, после чего обновляется информация об определенной задаче по её ID, указанному в таблице.

На рисунке 2.3 представлена детальная схема взаимодействия между модулем интеграции и системой YouTrack.

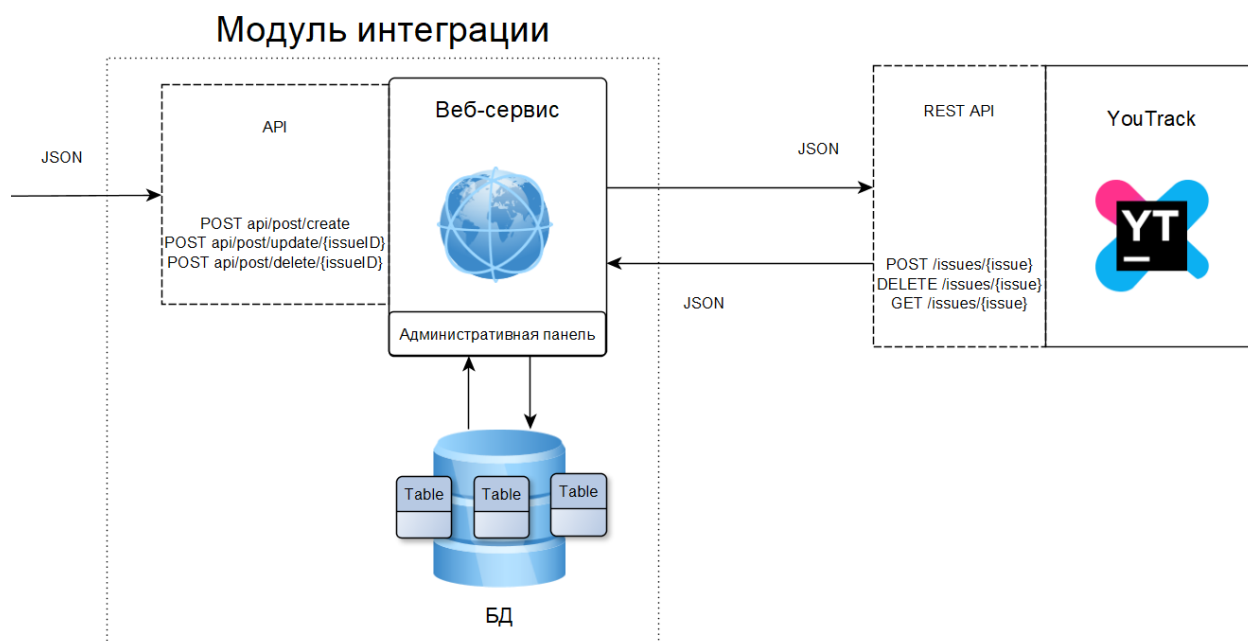


Рисунок 2.3 – Детальная схема взаимодействия модуля интеграции и YouTrack

На схеме указаны некоторые разработанные методы, позволяющие использовать REST API YouTrack для передачи данных. Данные в веб-сервис от модуля расширения XWiki приходят в формате JSON. Обмен данными с системой YouTrack также осуществляется в формате JSON.

2.3 Построение функциональной диаграммы с декомпозицией

Разработанная система предназначена ведения учета, анализа и передачи поступивших данных в систему трекинга задач YouTrack.

Система имеет следующие функции:

- 1) Учет информации о требованиях;
- 2) Учет информации о пользователях;
- 3) Учет информации о событиях;
- 4) Сравнение данных;
- 5) Формирование запроса к системе YouTrack.

Веб-сервис работает в автоматическом режиме, все переданные данные анализируются и передаются в систему YouTrack. Данный процесс позволит существенно сократить время на внесение низкоуровневых требований в систему YouTrack и на обновление этих данных напрямую из вики-системы XWiki.

На основе функций системы был смоделирован процесс интеграции данных при помощи нотации функционального моделирования IDEF0.

IDEF0-диаграмма показана на рисунке 2.4.

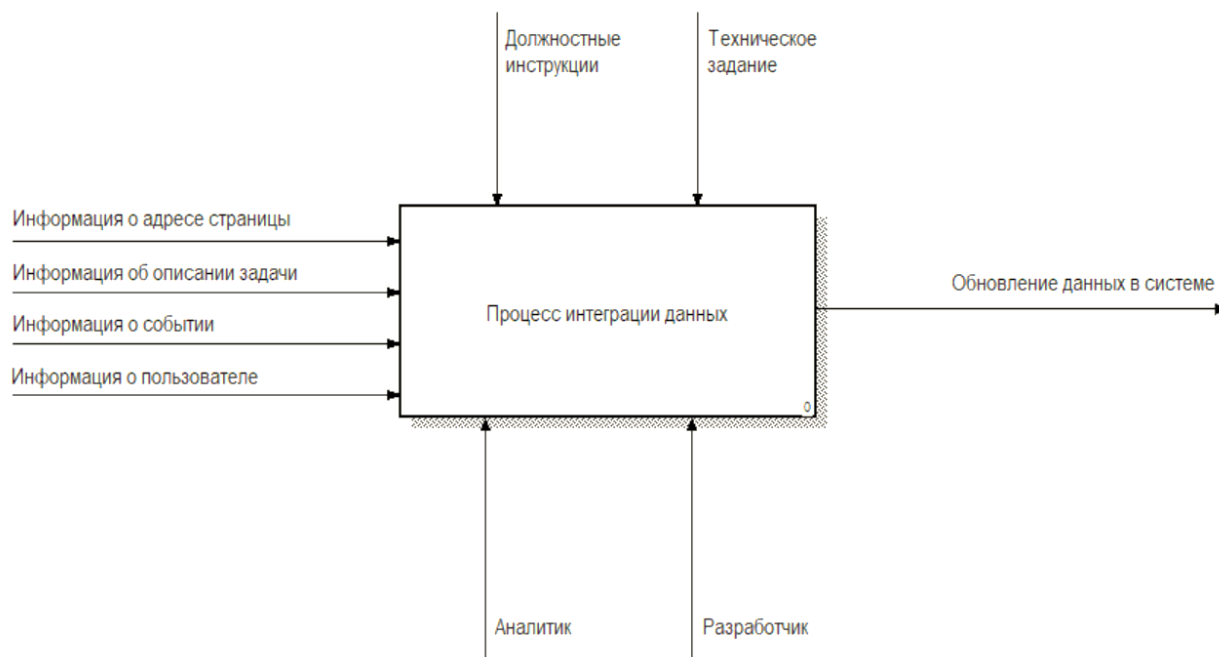


Рисунок 2.4 – IDEF0-диаграмма ИС

Весь процесс интеграции данных можно декомпозировать на подсистемы. Декомпозиция процесса интеграции данных представлена на рисунке 2.5.



Рисунок 2.5 – Декомпозиция процесса интеграции данных

Подсистема «Анализ данных на веб-сервисе» является основной подсистемой модуля интеграции. Создание функциональной модели подсистемы дает возможность точно закрепить, какие информационные объекты используются при выполнении деловых процессов и отдельных операций.

Функциональная диаграмма показана на рисунке 2.6.

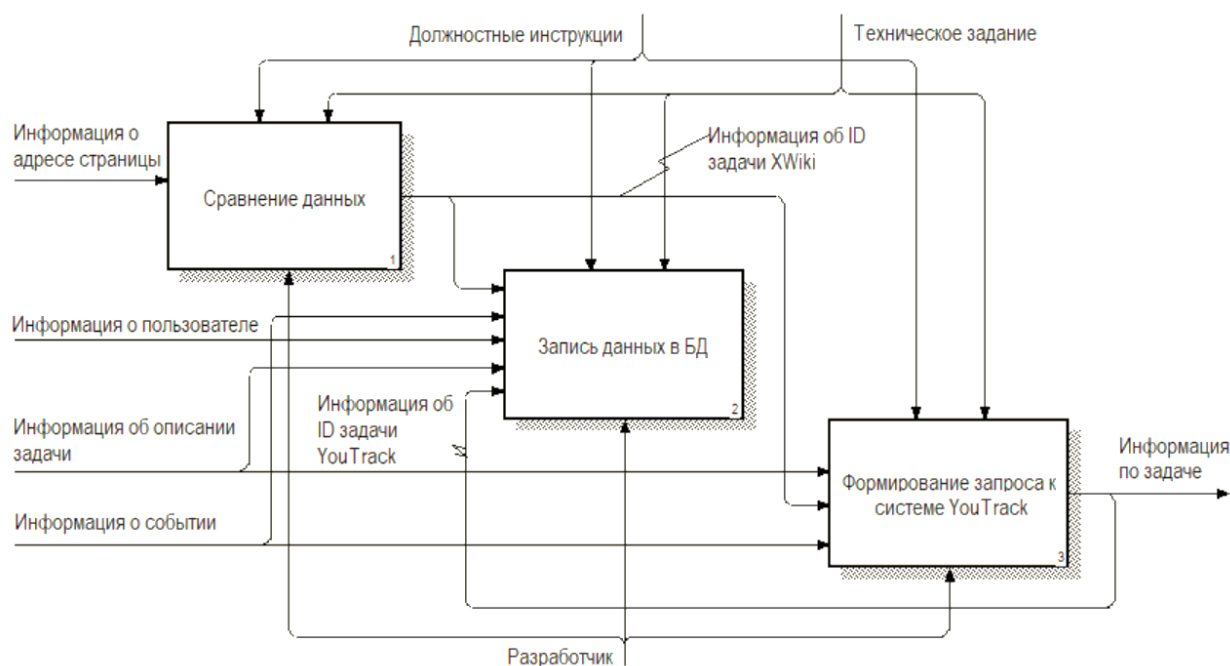


Рисунок 2.6 – Функциональная диаграмма

Функция «Сравнение данных» позволяет соотнести ID задачи XWiki с ID задачи в YouTrack. В случае, если такая пара в таблице находится, то по поступившим данным происходит обновление задачи в системе YouTrack. В случае, если соотнести ID не удалось, добавляется новая запись в базу данных. На основании ID задачи XWiki создается задача в YouTrack. ID создаваемой задачи в YouTrack фиксируется в таблице соотношений в БД модуля интеграции.

Функция «Запись данных в БД» обеспечивает учет и хранение информации о пользователях, соотношений ID задач двух систем и событиях, происшедших в системе XWiki. Информация о событиях необходима для

функционирования подсистемы журналирования, в которой отражается вся поступившая информация.

Функция «Формирование запроса к системе YouTrack» обеспечивает формирование запроса к системе YouTrack на основании всех полученных данных от системы XWiki. Запрос может быть сформирован как на обновление, добавление и удаление данных в системе YouTrack.

2.4 Проектирование реляционной модели базы данных

Реляционная модель является наиболее простой и привычной формой представления данных в виде таблиц.

Главное достоинство реляционной модели – это простота инструментальных средств ее поддержки. К недостаткам можно отнести жесткость структуры данных и прямую зависимость скорости ее работы от размера базы данных. Однако, просмотр всей базы данных для многих операций, представленных в такой модели, может оказаться необходимым.

Для организации базы данных будем использовать реляционную СУБД. Потому необходимо, чтобы была разработана логическая структура реляционной базы данных, так как на её основе будет осуществляться решение задачи. Используем процессный подход к разработке базы данных, определяя состав только тех данных, которые необходимы для решения задачи.

Информация в базе данных представлена четырьмя отношениями, которые описаны в таблице 2.3.

Таблица 2.3 – Учетная информация

Отношение	Атрибут	Описание
Пользователи	Код	Код пользователя
	Имя	Имя пользователя
Журналы	Код	Код журнала
	Соотношение	Код соотношения

Продолжение таблицы 2.3

	Пользователь	Код пользователя
	Дата	Дата записи
	Событие	Наименование события
Соотношения	Код	Код соотношения
	Идентификатор XWiki	Уникальный ключ требования XWiki
	Идентификатор YouTrack	Уникальный ключ требования YouTrack
Настройки	Код	Код настройки
	Адрес YouTrack	Адрес соединения с YouTrack
	E-mail	Адрес электронной почты, по которому идет аутентификация при взаимодействии с YouTrack
	Пароль YouTrack	Пароль, по которому идет аутентификация при взаимодействии с YouTrack
	Логин	Логин для входа в административную панель
	Пароль	Пароль для входа в административную панель

Модель данных представляет собой схему данных предметной области, создающаяся для правильного отображения действительности в ИС. Данные моделируются с целью построения базы данных.

Концептуальный уровень разрабатываемой БД является обобщающим представлением данных. Концептуальная модель предметной области описывает логическую структуру данных. Она является наиболее полным представлением требований к данным со стороны пользователей информационной системы. Все сущности, их атрибуты и связи предметной области представлены в концептуальной модели. Диаграмма сущность-связь (ERD) представлена в приложении Б.

На уровне ключей (KB – level), помимо имен сущностей и связей, показаны первичные, альтернативные и внешние ключи сущностей. Указываются также специфицированные свойства связей (их кардинальность и идентифицируемость).

Диаграмма KB-уровня представляет логическую структуру связей сущностей, которые составляют предметную область деятельности. Для описываемой предметной области концептуальная модель на уровне ключей представлена в приложении В.

На уровне атрибутов (FA-level) показаны все атрибуты сущностей. Данная диаграмма включает наиболее полные определения структуры разрабатываемой системы. Концептуальная модель на уровне атрибутов для данной предметной области представлена в приложении Г.

2.5 Разработка спецификации требований к системе

2.5.1 Требования к системе

Таблица 2.4 – Группы требований

Символ	Группа требований
F	Общие функциональные требования

Продолжение таблицы 2.4

P	Требования к средствам интеграции
IS	Требования к информационной безопасности
TS	Требования к техническому обеспечению
SR	Требования к программному обеспечению
RG	Требования к надежности
D	Требования к документации

2.5.2 [F] Общие функциональные требования

Таблица 2.5 – Общие функциональные требования

Код требования	Требования	Примечания
F.01	Требования к обработке данных	
F.01.01	Система должна добавлять ссылку на страницу с требованием XWiki в описании требования YouTrack	
F.01.02	Система должна сравнивать полученные данные из системы XWiki с имеющимися данными в YouTrack	
F.02	Требования к передаче данных	
F.02.01	Система должна обеспечивать возможность отправки данных в YouTrack	
F.02.02	Система должна обеспечивать возможность принятия данных из XWiki	

Продолжение таблицы 2.5

F.02.03	Система должна обеспечивать возможность принятия данных из YouTrack	
F.03	Общие требования	
F.03.01	Система должна создавать задачу в YouTrack на основе полученных данных из XWiki	
F.03.02	Система должна обновлять описание задачи YouTrack на основе полученных данных из XWiki	
F.03.03	Система должна обеспечивать предоставление своего функционала другим информационным системам посредством API	
F.03.04	Система должна заполнять остальные поля по умолчанию при создании задачи в YouTrack	

2.5.3 [P] Требования к средствам интеграции

Таблица 2.6 – Требования к средствам интеграции

Код требования	Требования	Примечания
P.01	Требования к интеграции с YouTrack	
P.01.01	Система должна обеспечивать интеграцию веб-сервиса с YouTrack в целях обеспечения обмена данными	

Продолжение таблицы 2.6

P.01.02	Система должна осуществлять взаимодействие веб-сервиса и YouTrack через REST API YouTrack	
P.02	Требования к интеграции с XWiki	
P.02.01	Система должна обеспечивать интеграцию веб-сервиса с XWiki в целях обеспечения обмена данными	
P.02.02	Система должна осуществлять взаимодействие модуля интеграции и XWiki через программные средства модуля расширения XWiki и API веб-сервиса	

2.5.4 [IS] Требования к информационной безопасности

Таблица 2.7 – Требования к информационной безопасности

Код требования	Требования	Примечания
IS.01	Система должна поддерживать проведение комплекса мероприятий по обеспечению информационной безопасности	Защиты от несанкционированного доступа, нарушения целостности и достоверности хранящейся и передаваемой информации
IS.02	В системе должна обеспечиваться конфиденциальность данных	

2.5.5 [TS] Требования к техническому обеспечению

Таблица 2.8 – Требования к техническому обеспечению

Код требования	Требования	Примечания
TS.01	Требования для Web-сервера	
TS.01.01	16-ядерный процессор;	
TS.01.02	Оперативная память: 8 ГБ	
TS.01.03	Дисковое пространство: жесткий диск с 512 ГБ	
TS.01.04	Канал связи между серверами: 1 Гбит/с	
TS.01.05	Внешний канал связи: 100 Мбит/с	

2.5.6 [SR] Требования к программному обеспечению

Таблица 2.9 – Требования к программному обеспечению

Код требования	Требования	Примечания
SR.01	Требования к операционным системам	
SR.01.01	Система должна поддерживать работу в ОС Microsoft Windows, Unix (Linux), Apple MacOS	
SR.02	Требования к браузеру	
SR.02.01	Internet Explorer 8 и выше (только для Windows); Mozilla Firefox 13 и выше; Safari 5 и выше; Google Chrome 15 и выше; Opera 11 и выше.	

2.5.7 [RG] Требования к надежности

Таблица 2.10 – Требования к надежности

Код требования	Требования	Примечания
RG.01	Программное обеспечение системы должно поддерживать восстановление своего функционирования при перезапуске аппаратных средств.	
RG.02	Система должна обеспечить повышенный уровень сохранности информации при авариях, отказах технических средств (в том числе – потеря питания)	
RG.03	Система должна обеспечивать сохранность и целостность ранее введенных данных при обновлениях	

2.5.8 [D] Требования к документации

Таблица 2.11 – Требования к документации

Код требования	Требования	Примечания
D.01	Разработка документации по API системы должна осуществляться в соответствии с настоящим Техническим требованием	В состав технических требований входят: Документация по API системы; Дистрибутивы сервера

Продолжение таблицы 2.11

D.02	Документация должна быть оформлена в текстовом формате на русском языке и представлена Заказчику в электронном виде	
------	---	--

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДУЛЯ ИНТЕГРАЦИИ

3.1 Обзор шаблонов проектирования архитектур системы

Шаблон проектирования – это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

MVC (Model-View-Controller) — это фундаментальный шаблон проектирования, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь разработчикам.

MVC состоит из трех частей, которые и дали ему название:

- 1) Model (Модель)
- 2) View (Вид или представление)
- 3) Controller (Контроллер).

Под моделью, обычно понимается часть, которая содержит в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то, как они отображаются, не трогая саму модель.

В обязанности представления входит отображение данных полученных от модели. Однако представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным.

Контроллер перехватывает событие извне и в соответствии с заложенной в него логикой, реагирует на это событие, изменяя модель посредством вызова соответствующего метода. После изменения модель использует событие о том, что она изменилась, и все подписанные на это события представления, получив его, обращаются к модели за обновленными данными, после чего их и отображают.

Шаблон проектирования MVC представлен на рисунке 3.1.

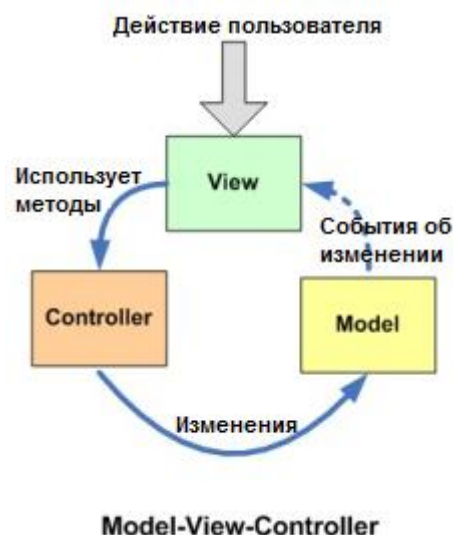


Рисунок 3.1 – Схема шаблона проектирования MVC

Существует множество различных шаблонов проектирования, в том числе и производных от MVC. Рассмотрим такие шаблоны проектирования как MVP и MVVM.

Шаблон проектирования MVP (Model-View-Presenter) позволяет создавать абстракцию представления. Для этого необходимо выделить интерфейс представления с определенным набором свойств и методов. Презентер, в свою очередь, получает ссылку на реализацию интерфейса, подписывается на события представления и по запросу изменяет модель.

Схема, иллюстрирующая данный подход представлена на рисунке 3.2.

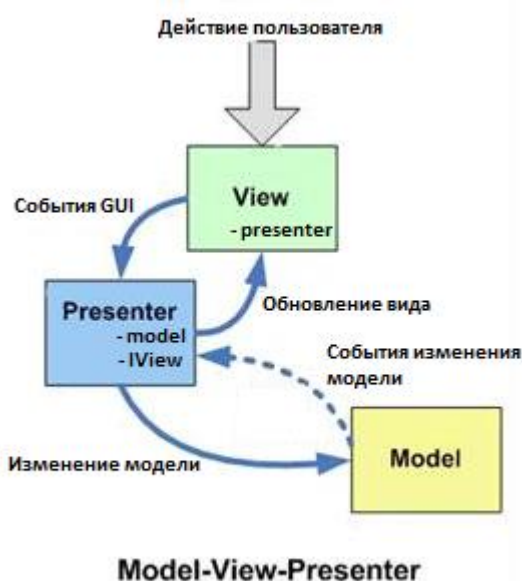


Рисунок 3.2 – Схема шаблона проектирования MVP

Каждое представление должно реализовывать соответствующий интерфейс. Интерфейс представления определяет набор функций и событий, необходимых для взаимодействия с пользователем. Презентер должен иметь ссылку на реализацию соответствующего интерфейса, которую обычно передают в конструкторе. Логика представления должна иметь ссылку на экземпляр презентера. Все события представления передаются для обработки в презентер и практически никогда не обрабатываются логикой представления.

Шаблон проектирования MVVM (Model-View-View Model) позволяет связывать элементы представления со свойствами и событиями View-модели. Можно утверждать, что каждый слой этого паттерна не знает о существовании другого слоя.

Схема шаблона проектирования MVVM представлена на рисунке 3.3.

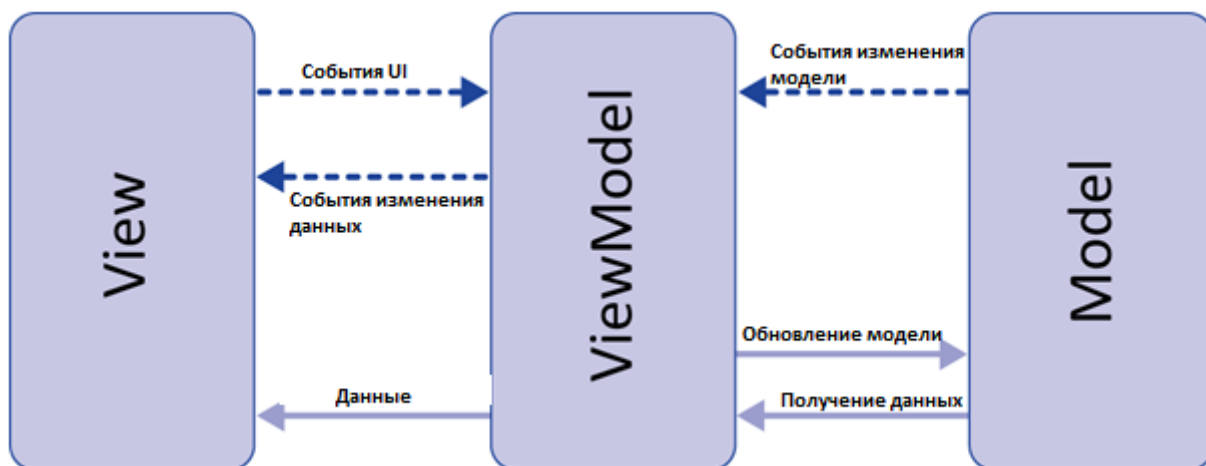


Рисунок 3.3 – Схема шаблона проектирования MVVM

При использовании этого шаблона проектирования, представление не реализует соответствующий интерфейс (IView). Представление должно иметь ссылку на источник данных (DataContext), которым в данном случае является View-Model. Элементы представления связаны (Bind) с соответствующими свойствами и событиями View-Model. В свою очередь, View-Model реализует специальный интерфейс, который используется для автоматического обновления элементов представления [7].

3.2 Архитектурный подход для организации веб-сервиса

При выборе архитектуры для организации веб-сервисов встает выбор между архитектурными подходами REST и SOAP. Рассмотрим по очереди каждый подход.

REST (Representational State Transfer) – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который, как правило, используется для построения веб-служб. Системы, поддерживающие REST, называются RESTful-системами. Схема стиля архитектуры REST представлена на рисунке 3.4.

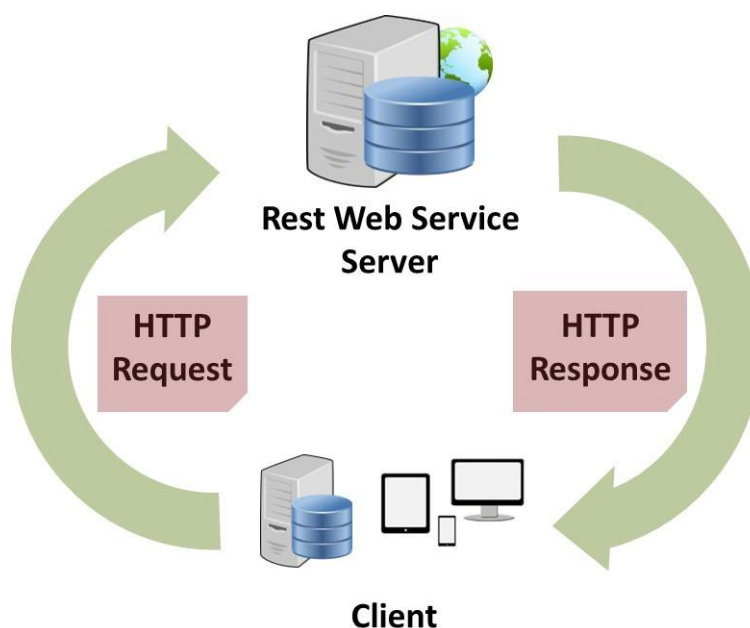


Рисунок 3.4 – Схема стиля архитектуры REST

В общем случае REST является очень простым интерфейсом управления информацией без использования каких-то дополнительных внутренних прослоек. Каждая единица информации однозначно определяется глобальным идентификатором, таким как URL. Каждая URL имеет строго заданный формат.

За весь функционал по управлению информацией сервиса отвечает протокол передачи данных HTTP. HTTP управляет данными с помощью четырех методов:

- GET (получить);

- PUT (добавить, обновить);
- POST (добавить, изменить, удалить);
- DELETE (удалить).

Таким образом, действия CRUD (Create-Read-Update-Delete) могут выполняться как со всеми четырьмя методами, так и только с помощью GET и POST [8].

SOAP (Simple Object Access Protocol) – это протокол обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC). Сейчас протокол используется для обмена произвольными сообщениями в формате XML.

SOAP использует целое семейство протоколов и стандартов, что на практике означает, что это более тяжеловесный и сложный вариант с точки зрения машинной обработки. Из-за этого он работает медленнее.

Так как SOAP работает с операциями, а REST с ресурсами, целесообразно использовать SOAP при реализации сервисов, использующие транзакции и сложную логику.

В рамках реализации модуля интеграции для взаимодействия двух систем в нашем случае рациональнее использовать REST-подход в создании веб-сервиса, отличающийся простотой и легкостью при пакетной передаче данных [9].

3.3 Выбор языка программирования и среды разработки

Для выбора языка программирования был проведен сравнительный анализ популярных языков веб-программирования.

Для сравнения были выбраны языки Ruby, PHP и Python.

Ниже представлена таблица 3.1 сравнения языков программирования, в которой представлены отличительные особенности каждого из языков по некоторым важным критериям.

Таблица 3.1 – Сравнение языков программирования

Критерии	Ruby	PHP	Python
Доля рынка (в качестве серверного языка)	0,6%	82,5%	0,2%
Простота освоения	Низкая	Средняя	Высокая
Производительность	Высокая	Высокая	Низкая
Удобство в использовании	Относится к языкам программирования сверхвысокого уровня, то есть обладает высоким уровнем абстракции и предметным подходом в реализации алгоритмов и работает на множестве платформ	PHP следует классическому подходу и тщательно документирован	Особенности синтаксиса стимулируют программиста писать хорошо читаемый код

На сегодняшний день PHP является самым популярным языком для WEB-разработчиков. На нем пишут CMS и дополнения к серверной части сайтов. На PHP сегодня работает более 2/3 всех интернет-ресурсов, от небольших визиток до крупнейших социальных сетей. В PHP нет жесткой структуры. Он гибок и прост, исходный код открытый.

Исходя из проведенного анализа, для разработки веб-сервиса был выбран язык PHP.

В качестве среды разработки была выбрана среда PhpStorm. Данная IDE поставляется бесплатно для учителей и студентов профильных ВУЗов. Она идеально подойдет для полноценной full-stack веб-разработки. Также поддерживает фреймворки Symfony, Zend, Yii2, CakePHP, основные CMS системы Magento, Drupal, Wordpress, отладчики Zend Debugger и Xdebug, а для фронтенда доступны CSS3, HTML5, Sass, CoffeeScript, TypeScript, Stylus, Less и другие инструменты [10].

3.4 Выбор инструментов для разработки

В разработке веб-ресурсов выбор инструмента критически важен, т.к. именно этот выбор предопределяет скорость, стоимость, расширяемость и масштабируемость продукта в целом.

Для разработки качественного продукта уже существует огромное количество хороших инструментов, облегчающих процесс создания, проектирования, и сопровождения проекта. Технологии в веб-среде меняются очень быстро, развитие не происходит в одностороннем порядке, изменения также касаются способов и методов потребления информации.

В современном мире многие веб-разработчики используют готовые системы управления контентом (Content Management System) с целью оптимизации своих расходов. При разработке типовых шаблонных сайтов данный подход оправдывает себя т.к. CMS это предустановленный набор модулей, полностью реализованная архитектура, которая является очень ограниченной для сторонней модификации разработчиками [11].

Системы управления контентом обязаны быть универсальными, цена этого: избыточность кода, более сложная архитектура, меньшая гибкость [12].

Фреймворк представляет собой программный каркас, реализующий базовую архитектуру проекта.

Критерии выбора фреймворка:

- Низкий порог вхождения в работу с фреймворком (фреймворк предлагает новичкам комфортную среду для быстрого старта);
- Русскоязычное сообщество (возможность быстро получить помощь или найти интересующий материал на русском языке);
- Производительность (производительность измеряется в количестве операций обработанных в течение секунды);
- Занимаемая память (характеристика (в мегабайтах) отвечает за количество занимаемой оперативной памяти фреймворком при выполнении);

- Время выполнения (время, затрачиваемое системой для выполнения задачи. Измеряется от начала выполнения задачи до выдачи результата системой (количество ожидания времени, до получения ответа от сервера));
- Количество подключаемых файлов (характеристика, отвечающая за количество подключаемых файлов. Меньшее количество подключаемых файлов обеспечивает быстрый запуск ресурса);
- Возможность установки на любой хост-сервис (возможность установки на любой хост-сервис без использования привилегированных прав доступа к хост-сервису).

Критерии выбора фреймворков обусловлены современными требованиями к разработке веб-ресурсов, так же наличием личного опыта работы на языке php и отведенными временными рамками на разработку веб-ресурса.

Для сравнения платформ веб-разработки были выбраны современные фреймворки, широко применяемые отечественными разработчиками на рынке разработки веб-ресурсов:

- Yii 2;
- Laravel 5;
- Zend Framework 3.

Yii PHP Framework, основанный на компонентной структуре, для разработки масштабных веб-приложений. Позволяет максимально применить концепцию повторного использования кода, существенно ускоряет процесс веб-разработки. Кроссплатформенный фреймворк написан на языке PHP, реализует парадигму MVC.

Laravel PHP Framework с открытым кодом (open source). Кроссплатформенный Фреймворк написан на языке PHP, имеющий интегрированную систему модульного тестирования, реализует парадигму MVC.

Zend Framework PHP Framework с открытым кодом (open source) для разработки веб-приложений, разрабатываемый компанией Zend. Фреймворк Zend объединяет в себе прогрессивные методы программирования, удобную схему размещения файлов, встроенную поддержку для решения многочисленных задач возникающих во время разработки. Zend Framework кроссплатформенный, написан на языке PHP, реализует парадигму MVC.

В таблице 3.2 представлен сравнительный анализ трех популярных PHP-фреймворков для разработки веб-сервиса.

Таблица 3.2 – Сравнение PHP-фреймворков

Критерии	<u>Yii 2</u>	<u>Laravel 5</u>	<u>Zend Framework 3</u>
Уровень вхождения (необходимый уровень знаний для работы)	PHP5, ООП.	ООП, PHP, консоль.	PHP5, ООП, шаблоны проектирования.
Русскоязычное сообщество	Есть	Есть	Есть
Производительность (количество обработанных запросов в 1 сек.)	2664.61	600.42	1197.26
Занимаемая память (количество занимаемой оперативной памяти фреймворком при выполнении задачи, Мб)	0.91	1.49	1.08
Время, затрачиваемое системой для выполнения задачи. Измеряется в миллисекундах от начала выполнения задачи до выдачи результата системой.	1.639	3.940	2.989
Количество подключаемых файлов. Меньшее количество подключаемых файлов обеспечивает быстрый старт системы.	46	176	188
Возможность установки на любой хост-сервис	Да	Да	Да

В таблице представлены данные, полученные на основании проведенного тестирования компанией «NIX Solutions», опубликованные в статье «Как выбрать PHP-фреймворк» [13].

Сформулированным требованиям из представленных фреймворков соответствует только Yii2, так как он имеет хорошую производительность, огромное количество хорошей русскоязычной документации, включая примеры в глобальной сети Интернет.

Yii2 использует шаблон проектирования MVC, который широко применяется в веб-программировании. Помимо этого, Yii использует фронт-контроллер, называемый приложением (application), который инкапсулирует контекст обработки запроса. Приложение собирает информацию о запросе и передает её для дальнейшей обработки соответствующему контроллеру [14].

На рисунке 3.5 статическая структура приложения Yii.

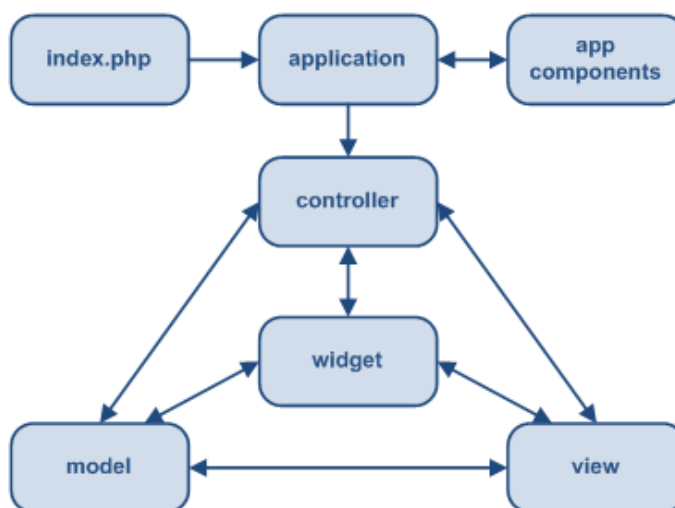


Рисунок 3.5 – Статическая структура приложения Yii

Схема, представленная на рисунке 3.6, описывает типичную последовательность процесса обработки пользовательского запроса приложением.

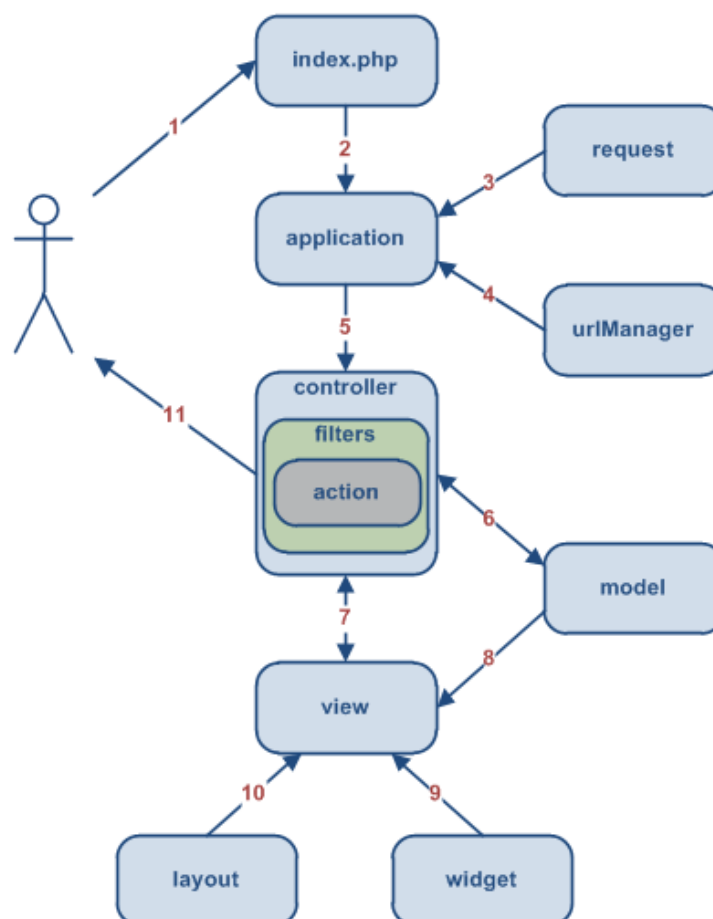


Рисунок 3.6 – Типичная последовательность работы приложения Yii

В целях сокращения временных затрат на разработку веб-сервиса, дополнительно к фреймворку применяется библиотека Guzzle. Это простой HTTP-клиент для PHP, который помогает автоматизировать осуществляемые запросы к сторонним сервисам, а также их настройку, используя cURL или другие библиотеки для передачи данных по различным протоколам [15].

3.5 API системы YouTrack

YouTrack REST API позволяет программно выполнять различные действия в трекере:

- Импорт задач из какой-либо системы трекинга задач в YouTrack для плавного перехода;

– Создание, изменение, удаление и выполнение других операций с различными сущностями;

– Управление проектами, пользователями, группами, ролями, типами ссылок на задачи и пользовательскими атрибутами.

REST API возвращает и принимает данные в формате JSON. Он работает всегда. Администратор может указать, каким сайтам или источникам разрешен доступ к YouTrack через REST.

Взаимодействие с REST API YouTrack осуществляется через протокол HTTP, и для этого можно использовать любой клиент, который может работать по этому протоколу. Кроме того, для взаимодействия с RESTful API так же может использоваться веб-браузер.

API-интерфейс YouTrack поддерживает два типа аутентификации:

- HTTP BASIC Auth – предоставление своих учетных данных, используя протокол HTTP.
- OAuth 2.0 – предоставление прав определенному веб-приложению на доступ к REST API YouTrack.

Веб-сервис использует HTTP Basic Auth тип аутентификации, по которому передает E-mail и пароль.

При отправке данных на веб-сервис происходит генерация уникального ключа. Ключ генерируются на основе данных предназначенных для передачи. Эти данные шифруются библиотекой mcrypt по сгенерированному ключу и инициализирующему вектору. Инициализирующий вектор представляет собой блок случайных данных сгенерированных библиотекой mcrypt и предназначен для уникальности шифра при каждой передаче данных. В полученной зашифрованной строке происходит попарная перестановка символов для увеличения степени защищенности передаваемых данных. Преобразованная строка кодируется библиотекой base64 для передачи по протоколу HTTP. Полученный в результате шифр является ключом для валидации передаваемых данных на веб-сервисе. Данный способ шифрования является двусторонним, подлежит шифрованию и дешифрованию.

Основные методы API, используемые при передаче данных:

- GET /api/issues/{issueID} - этот метод используется для получения информации по задаче;

- POST /api/issues - этот метод используется для создания новой задачи.

Для внесения информации в новую задачу необходимо отправить JSON:

```
{
  "summary": "Houston!",
  "description": "We have a problem!",
  "project": {
    "id": "0-0"
  }
}
```

Где «summary» – название задачи, «description» – описание задачи, «id» – номер проекта;

- POST /api/issues/{issueID} - этот метод используется для обновления информации по задаче. Необходимо отправить JSON с полями, которые следует обновить;

- DELETE /api/issues/{issueID} - этот метод используется для удаления задачи;

- GET /api/admin/projects/{projectID} – этот метод используется для получения информации о проекте;

- POST /api/admin/projects – этот метод используется для создания нового проекта. Для внесения информации в новый проект необходимо отправить JSON:

```
{
  "shortName": "PTR",
  "name": "Project TR",
  "leader": {
    "id": "1-1"
  }
}
```

Где «shortName» – краткое название проекта, «name» – полное название проекта, «id» – номер пользователя, являющегося лидером проекта.

3.6 Описание разработанных методов системы

Веб-сервис работает на фреймворке Yii2 и использует шаблон проектирования системы MVC. На вход реализованным методам веб-сервиса поступает пакет JSON следующей структуры, представленной на рисунке 3.7.

```
{
  "header": "название страницы",
  "content": "контент",
  "url": "https://name.xwiki.com/xwiki/bin/View/Main/Str",
  "user": "sergey",
  "key": "0L3QsNC30LLQsNC90LjQtSDRgdGC0YDQsNC90LjRhtGL0LrQvtC90YLQtdC90YI=/aHR0cHM6Ly9uYW11Lnh3aWtpLmNvbS94d2lraS9iaW4vVmlldy9NYW1uL1N0cg=="
}
```

Рисунок 3.7 – Схема JSON

Структура JSON имеет следующие ключи:

- «header» содержит в себе название страницы XWiki, одновременно являясь её идентификатором ID;
- «content» содержит информацию на странице;
- «url» содержит в себе полный адрес страницы XWiki;
- «user» содержит имя пользователя;
- «key» содержит в себе уникальный секретный ключ. Он представляет собой часть конкатенированной передаваемой информации, которая была зашифрована.

Сервис способен принимать запросы POST для добавления, редактирования и удаления задач.

Для этого в системе предусмотрен ApiController, который имеет следующие методы:

- POST api/post/create – метод, позволяющий создать новую задачу.

Алгоритм работы метода представлен в приложении Д. Веб-сервис принимает JSON-пакет, по данным которого проводит поиск в базе данных существующих записей по задаче. В случае если их нет, будет создана новая задача в YouTrack исходя из полученной информации. В противном случае обновит задачу.

- POST `api/post/update/{issueID}` – метод, позволяющий обновить существующую задачу в YouTrack. В качестве параметра принимает ID задачи в YouTrack «issueID».

Алгоритм работы метода обновления задачи представлен в приложении Е.

- POST `api/post/delete/{issueID}` – метод, позволяющий удалить существующую задачу в YouTrack. В качестве параметра принимает ID задачи в YouTrack «issueID».

Алгоритм работы метода удаления задачи представлен в приложении Ж.

3.7 Особенности разработанного веб-сервиса

Веб-сервис представляет собой программное обеспечение, которое может располагаться на отдельном компьютере. Сервис устанавливает соединение с БД MySQL Server, где хранятся все данные логов и связи ID задач двух систем. Физическая модель данных представлена на рисунке 3.8.

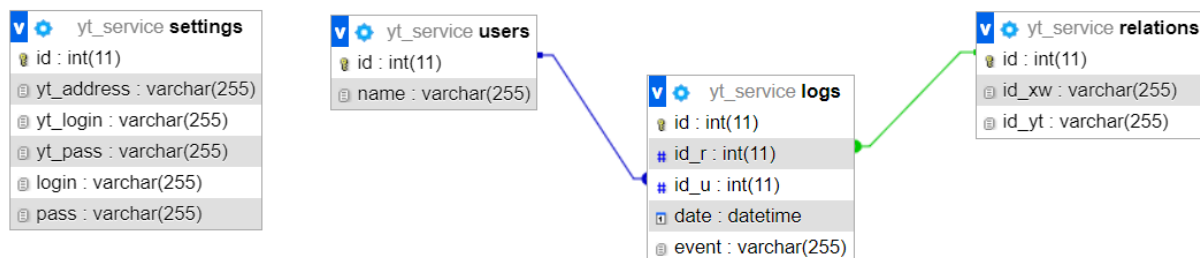


Рисунок 3.8 – Физическая модель базы данных веб-сервиса

Для просмотра журнала и внесения каких-либо настроек соединения веб-сервиса с системой трекинга задач YouTrack разработана административная панель. Маршрутизация в фреймворке Yii2 осуществляется менеджером URL Yii. Это компонент приложения, который используется для анализа входящих запросов, `parseRequest()`, а также для создания новых URL-адресов программным способом `createUrl()`. Запросы разбираются на маршруты, которые принимают форму «ControllerID/ActionID» [16].

На рисунке 3.9 представлена схема маршрутизации данных для вывода страницы.

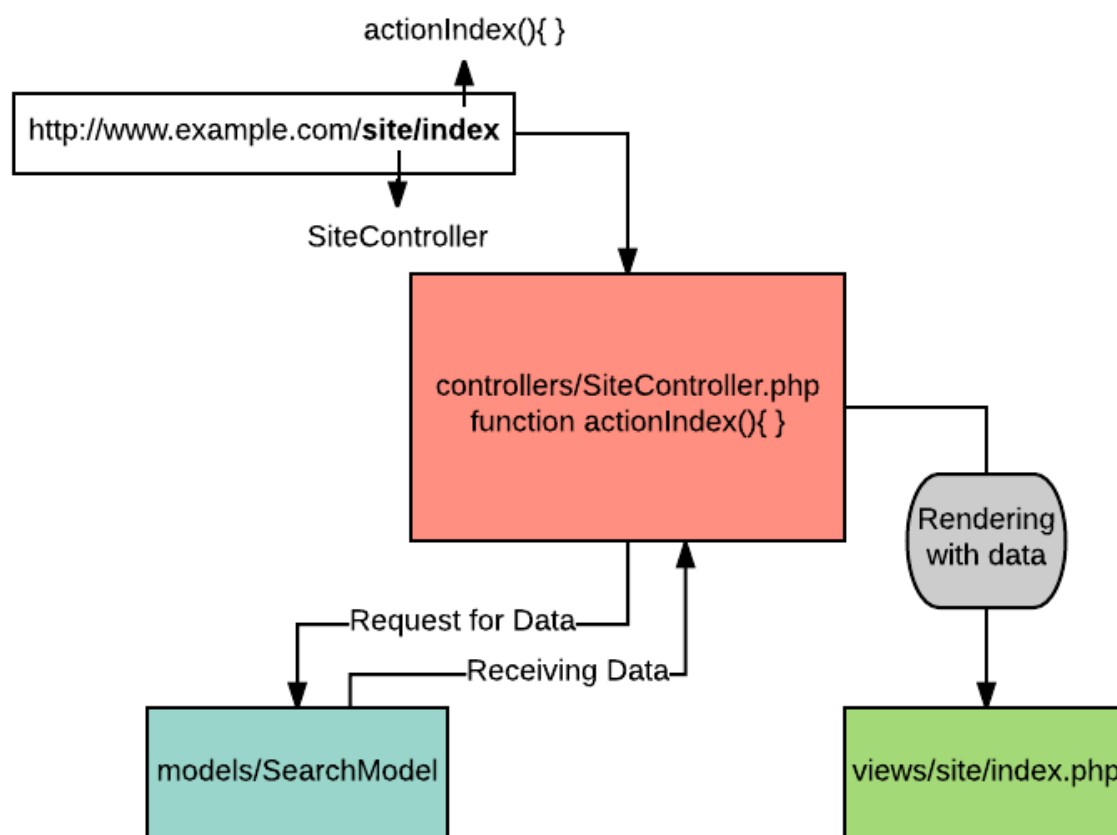


Рисунок 3.9 – Схема маршрутизации фреймворка Yii2

Менеджер URL-адресов фреймворка Yii, отвечает за то, какой контроллер и метод действий создавать и вызывать при поступлении запроса по URL-адресу от пользователя.

Работа с административной панелью осуществляется через контроллер AdminController, который имеет следующие методы:

- `actionIndex()` – метод, направляющий на главную страницу административной панели `admin/panel`, в случае если пользователь авторизован, и на страницу авторизации, если пользователь не авторизован. Страница авторизации представлена на рисунке 3.10.

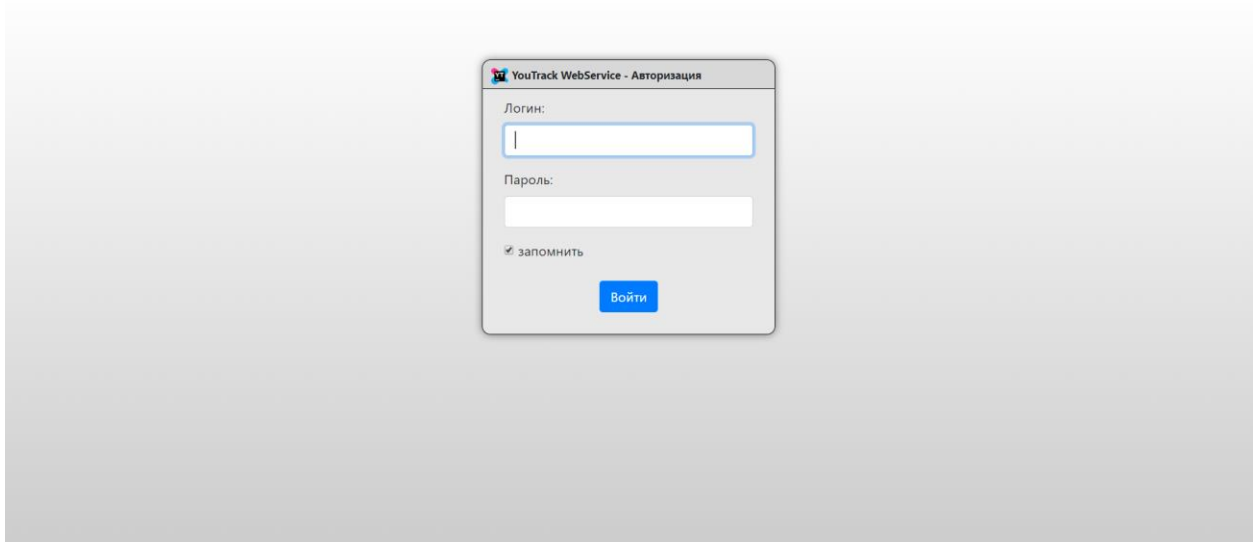


Рисунок 3.10 – Страница авторизации административной панели

- `actionPanel()` – метод, направляющий на страницу авторизации `admin/index`, если пользователь не авторизован, и на страницу административной панели, в случае если пользователь авторизован. Главная страница административной панели представлена на рисунке 3.11.

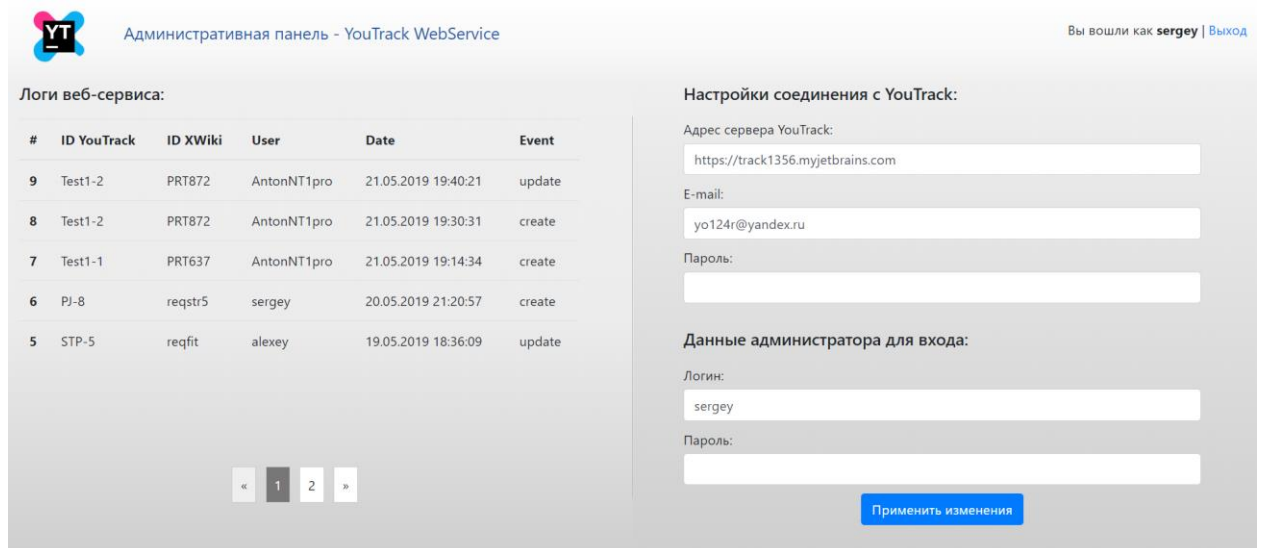


Рисунок 3.11 – Главная страница административной панели

- `actionLogout` – метод, позволяющий выйти из административной панели на страницу авторизации.

При отсутствии соединения с системой YouTrack пользователю будет выброшено исключение, информирующее об отсутствии соединения. При

валидации уникального ключа также может быть выброшено исключение, в случае если ключ не пройдет валидацию.

В системе трекинга задач YouTrack создаются новые задачи и новые проекты исходя из структуры принимаемого ключа «url» в JSON-пакете. На рисунке 3.12 видны созданные задачи из системы XWiki.

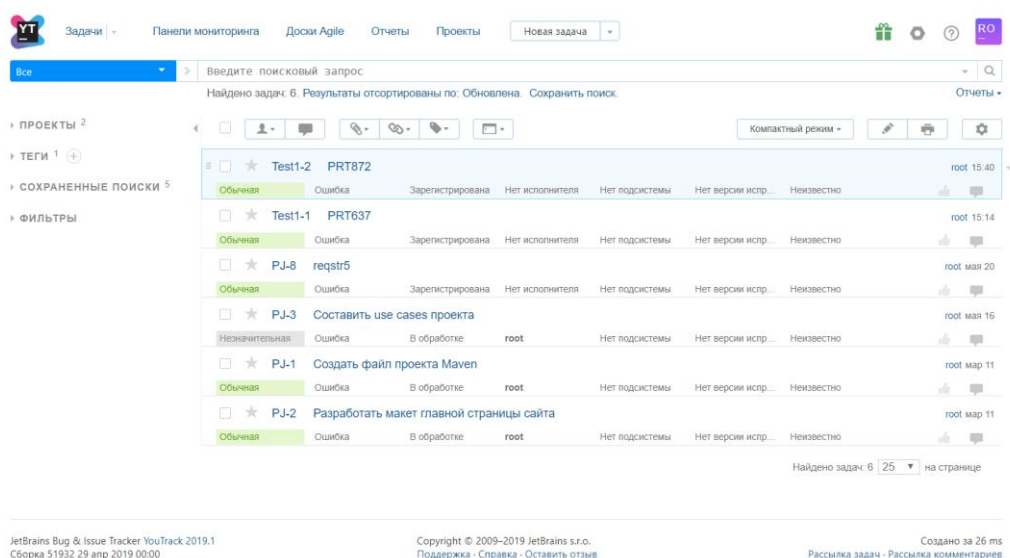


Рисунок 3.12 – Созданные задачи в системе YouTrack

В логах административной панели можно увидеть, что задача Test1-2 после создания обновлялась. В них отражаются все стандартные операции над задачами: создание, обновление и удаление. Записи логов не исчезают даже после удаления задачи.

4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТИ И РЕСУРСОСБЕРЕЖЕНИЕ

4.1 Предпроектный анализ

Данная выпускная квалификационная работа магистра направлена на разработку информационной системы с целью интеграции двух различных платформ посредством данной системы.

Целью данного раздела является определение перспективности и успешности научно-исследовательского проекта, разработка механизма управления и сопровождения конкретных проектных решений на этапе реализации.

Разработка данной информационной системы, представляющая собой модуль интеграции, позволит корпоративной базе знаний XWiki и системе трекинга задач YouTrack взаимодействовать между собой, автоматизируя процесс переноса требований из XWiki, конвертируя их в конкретные задачи для разработчиков.

Поскольку разработка является в большей степени практической, первичными потребителями являются коммерческие организации, разрабатывающие различное программное обеспечение и использующие такую систему управления задач как YouTrack.

4.1.1 Анализ конкурентных технических решений

Анализ конкурентных технических решений помогает вносить коррективы в научное исследование, чтобы успешнее противостоять конкурентам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

В качестве основных конкурентных технических решений были выбраны следующие разработки:

- интеграция YouTrack с XWiki, предлагаемая в данной работе (ф),
- интеграция JIRA с XWiki (1),

- интеграция Redmine с XWiki (2).

Результаты конкурентного анализа приведены в таблице 4.1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_i, \quad (1)$$

где K – конкурентоспособность научной разработки или конкурента;

B_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

Таблица 4.1 – Оценочная карта сравнения конкурентных интеграций

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		B_{ϕ}	B_{k1}	B_{k2}	K_{ϕ}	K_{k1}	K_{k2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Скорость работы	0,3	4	5	4	1,2	1,5	1,2
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,1	4	4	3	0,4	0,4	0,3
3. Надежность	0,05	4	4	3	0,2	0,2	0,15
4. Потребность в ресурсах памяти	0,05	5	4	4	0,25	0,2	0,2
5. Функциональная мощность (предоставляемые возможности)	0,1	4	5	4	0,4	0,5	0,4
6. Простота эксплуатации	0,05	5	4	2	0,25	0,2	0,1
Экономические критерии оценки эффективности							
7. Цена	0,2	4	4	5	0,8	0,8	1
8. Послепродажное обслуживание	0,05	4	4	3	0,2	0,2	0,15
9. Доступность	0,1	5	5	4	0,5	0,5	0,4
Итого	1	39	39	32	4,2	4,5	3,9

Критерий «Функциональная мощность» отражает наличие либо отсутствие дополнительных возможностей;

Под «Доступностью» понимается то, насколько открытой является интеграция. Например, интеграция Redmine с XWiki не столь прозрачна.

Под «Простотой эксплуатации» подразумевается то, насколько просто пользоваться системой, насколько ожидаемо её поведение в различных ситуациях.

Таким образом, сравнивая конкурентные интеграции по заданным критериям можно сделать вывод, что предлагаемая интеграция в данной работе выигрывает в простоте эксплуатации и меньшей потребности в ресурсах памяти.

4.1.2 Оценка готовности проекта к коммерциализации

На какой бы стадии жизненного цикла не находилась научная разработка, полезно оценить степень готовности представленной разработки к коммерциализации и выяснить уровень собственных знаний для ее проведения (или завершения).

Оценка готовности научного проекта к коммерциализации (или уровень имеющихся знаний у разработчика) определяется по формуле:

$$B_{\text{сум}} = \sum B_i \quad (2)$$

Таблица 4.2 – Бланк оценки степени готовности научного проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1.	Определен имеющийся научно-технический задел	4	4
2.	Определены перспективные направления коммерциализации научно-технического задела	5	4
3.	Определены отрасли и технологии (товары, услуги) для предложения на рынке	4	4
4.	Определена товарная форма научно-технического задела для представления на рынок	3	3
5.	Определены авторы и осуществлена	4	4

	охрана их прав		
6.	Проведена оценка стоимости интеллектуальной собственности	3	3
7.	Проведены маркетинговые исследования рынков сбыта	3	3
8.	Разработан бизнес-план коммерциализации научной разработки	2	2
9.	Определены пути продвижения научной разработки на рынок	2	3
10.	Разработана стратегия (форма) реализации научной разработки	4	4
11.	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	2	2
12.	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	2	3
13.	Проработаны вопросы финансирования коммерциализации научной разработки	2	2
14.	Имеется команда для коммерциализации научной разработки	1	1
15.	Проработан механизм реализации научного проекта	4	5
	Итого баллов	45	47

При проведении анализа по таблице, приведенной выше, по каждому показателю ставилась оценка по пятибалльной шкале. При этом система измерения по каждому направлению (степень проработанности научного проекта, уровень имеющихся знаний у разработчика) отличается. Так, при оценке степени проработанности научного проекта 1 балл означает не проработанность проекта, 2 балла – слабую проработанность, 3 балла – выполнено, но в качестве не уверен, 4 балла – выполнено качественно, 5 баллов – имеется положительное заключение независимого эксперта.

Для оценки уровня имеющихся знаний у разработчика система баллов принимает следующий вид: 1 – означает не знаком или мало знаю, 2 – в объеме теоретических знаний, 3 – знаю теорию и практические примеры применения, 4

– знаю теорию и самостоятельно выполняю, 5 – знаю теорию, выполняю и могу консультировать.

Таким образом, в ходе анализа получены сведения, подтверждающие, что разработка данного проекта достаточно перспективна (перспективность разработки выше среднего), исследование следует продолжать. Для получения большей результативности необходимо привлечение специалистов и экспертов.

4.1.3 Диаграмма Исикавы

Диаграмма причины-следствия Исикавы – это графический метод анализа и формирования причинно-следственных связей, инструментальное средство для систематического определения причин проблемы и последующего графического представления.

Построение диаграммы начинают с формулировки проблемной области/темы, которая является объектом анализа и наносится на центральную горизонтальную стрелку диаграммы. Затем выявляются факторы/группы факторов, влияющие на объект анализа.

Диаграмма Исикавы для интеграции двух систем представлена на рисунке 4.1.

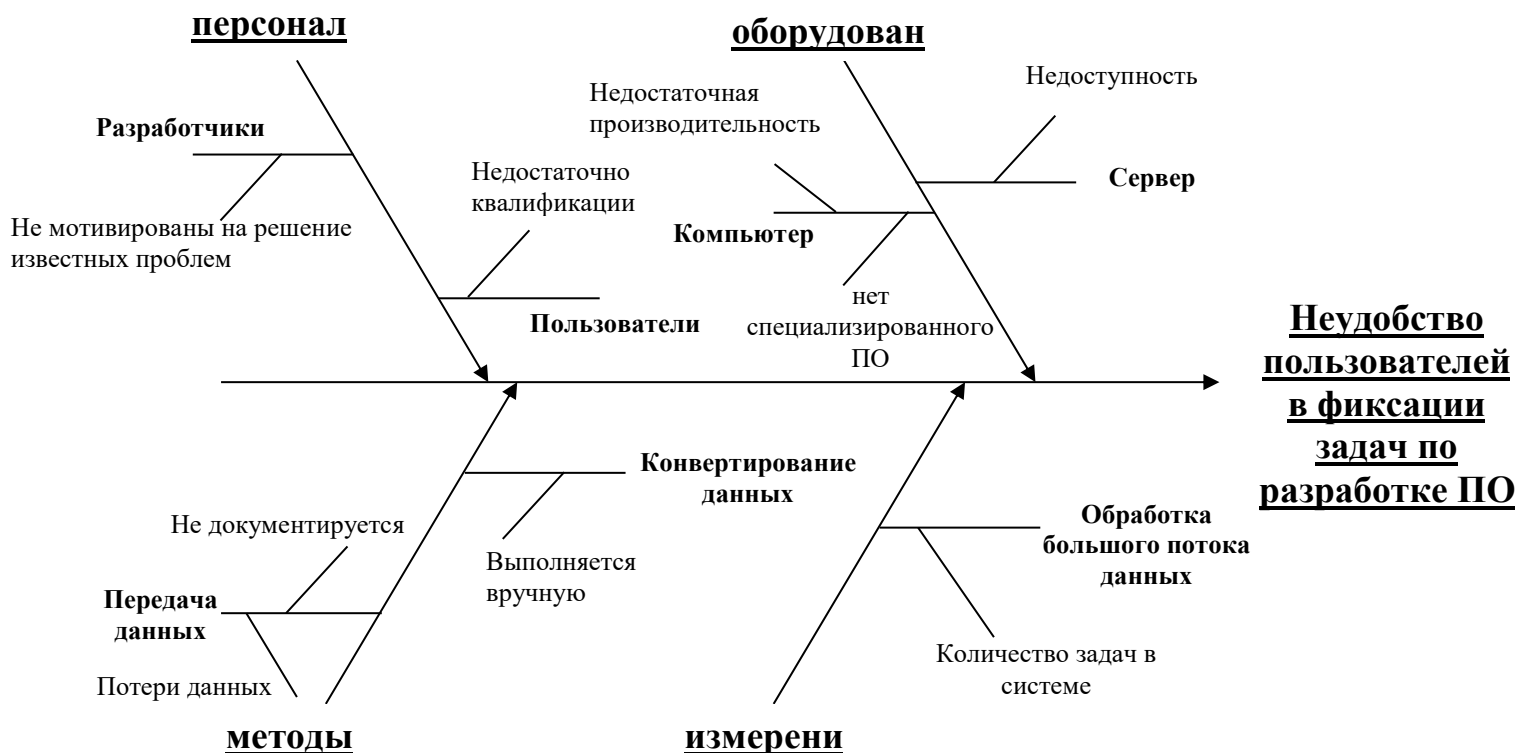


Рисунок 4.1 – Диаграмма Исикавы

По построенной диаграмме Исикавы можно сделать следующие выводы о наиболее влияющих факторах: для устранения проблемы неудобства пользователей в фиксировании задач по разработке ПО, необходимо мотивировать разработчиков на решении известных проблем; интегрировать системы с учетом пожеланий пользователей.

4.1.4 SWOT-анализ

Данный анализ представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта, его сильных и слабых сторон.

Первый этап анализа заключается в описании сильных и слабых сторон проекта, в выявлении возможностей и угроз для реализации проекта, которые проявились или могут появиться в его внешней среде.

Для проведения комплексного анализа информационной системы была составлена матрица SWOT для первого этапа, которая представлена в таблице 4.3.

Таблица 4.3 – Матрица SWOT-анализа проекта

	<p>Сильные стороны проекта: С1: Использование популярного формата обмена данными JSON; С2: Наличие подсистем, которых нет у аналогов; С3: Возможность легкой интеграции YouTrack с другими существующими системами; С4: Ясность и доступность для специалистов данного профиля. С5: Наличие контроля корректности и целостности данных.</p>	<p>Слабые стороны проекта: Сл1: Отсутствие финансирования; Сл2: Использование бесплатной БД; Сл3: Отсутствие большой команды по осуществлению проекта.</p>
<p>Возможности: В1: Рост спроса. В2: Создание команды, состоящей из опытных кадров. В3: Внедрение в сложную систему (применение на практике).</p>	<p>В1С2С3С5 Благодаря уникальным подсистемам, настраиваемой интеграции и наличию контроля целостности данных, система может выйти на более крупный рынок В2С1С4 Наличие квалифицированного разработчика и легкость освоения данной технологии позволяет создать команду из опытных кадров</p>	<p>В1В3Сл1Сл3 Отсутствие финансирования и большой команды могут воспрепятствовать выходу на крупный рынок и дальнейшему внедрению в сложную систему В1Сл2 Использование бесплатных баз данных может повысить риски нестабильной работы системы при внедрении её в более сложную и крупную систему, работающую с большими объемами данных В2Сл1 Из-за отсутствия финансирования создание команды из опытных специалистов невозможно</p>

<p>Угрозы: У1: Рост сложности проекта. У2: Появление достойных конкурентов. У3: Появление радикально новых технологий, исключающих применение разработки</p>	<p>У1С4 Благодаря простоте и прозрачности системы, её сложность не будет возрастать У2У3С2С4 Благодаря подсистемам, у которых нет аналогов, и доступности для специалистов данного профиля система может оставаться актуальной и конкурентоспособной</p>	<p>У1Сл2Сл3 Рост сложности проекта увеличит нагрузку на бесплатную базу данных и неблагоприятно отразится на команде разработчиков</p>
--	---	---

Ключевой проблемой проекта является отсутствие финансирования, из-за чего затруднительно активно внедрять и расширять данную систему. Рассмотрев слабые стороны проекта и возможности, можно прийти к выводу, что необходима стратегия поиска дополнительного финансирования проекта, которое позволит улучшать технологии разработанной системы, что положительно скажется на автоматизации бизнес-процессов предприятия, где система будет интегрирована.

Такие сильные стороны проекта, как наличие уникальных подсистем и возможность быстрой и легкой интеграции YouTrack с другими системами, позволят минимизировать угрозы, связанные с не востребованностью данной технологии и поиском дополнительного финансирования.

В итоге: для минимизирования угроз, в первую очередь, необходимо улучшать слабые стороны проекта. Как раз появление дополнительного денежного финансирования и высококвалифицированного персонала помогут преодолеть их.

4.2 Инициация проекта

4.2.1 Устав проекта

Устав проекта документирует бизнес-потребности, текущее понимание потребностей заказчика проекта, а также новый продукт, услугу или результат, который планируется создать.

Устав научного проекта должен иметь следующую структуру:

- Цели и результат проекта;
- Организационная структура проекта;
- Ограничения и допущения проекта.

Таблица 4.4 – Цели и результат проекта

Цели проекта	Разработка модуля интеграции между системой YouTrack и XWiki.
Ожидаемые результаты проекта	Взаимодействие системы YouTrack с системой XWiki через API.
Критерии приемки результата проекта	Соответствие всем пунктам ТЗ
Требования к результату проекта	Основным требованием к модулю интеграции является реализация создания задач, обновление и удаление их из системы YouTrack по поступившим запросам от системы XWiki

4.2.2 Ограничения и допущения проекта

Ограничения проекта – факторы, которые могут ограничить степени свободы участников команды проекта, а также «границы проекта» – параметры проекта, которые не будут реализованных в рамках данного проекта.

Таблица 4.5 – Ограничения проекта

Фактор	Ограничения / допущения
Бюджет проекта	400 тыс. р.
Сроки проекта	6 месяцев
Дата утверждения плана управления проектом	7 декабря 2018 года
Дата завершения проекта	1 июня 2019 года

Прочие ограничения и допущения	Отсутствие финансирования на начальном этапе
--------------------------------	--

4.2.3 Организация и планирование работ

Для организации оптимального процесса реализации проекта необходимо спланировать занятость каждого участника разработки, а также сроки проведения отдельных работ.

Этот этап служит для составления полного перечня проводимых работ, определения исполнителей для данных работ, а также формируется время продолжительности работы. Перечень приведен в таблице 4.6.

НР – научный руководитель, И – исполнитель.

Таблица 4.6 – Перечень работ и загрузка исполнителей

Этапы работы	Исполнители	Загрузки исполнителей
Постановка цели и задач	НР	НР – 100%
Подбор и изучение теоретического материала по тематике	НР, И	НР – 10% И – 90%
Разработка календарного плана	НР, И	НР – 80% И – 20%
Проектирование схемы интеграции	И	И – 100%
Проектирование базы данных	И	И – 100%
Реализация веб-сервиса	И	И – 100%
Разработка административной панели	И	И – 100%
Оформление пояснительной записки	И	И – 100%
Подведение итогов	НР, И	НР – 50% И – 50%

4.2.4 Продолжительность этапов работ

Для того чтобы определить ожидаемое значение продолжительности работ $t_{ож}$ применяется вероятностный метод – метод двух оценок t_{min} и t_{max} .

$$t_{ож} = \frac{3*t_{min}+2*t_{max}}{5}, \quad (2)$$

где t_{min} – минимальная трудоемкость работ, чел/дн;

t_{max} – максимальная трудоемкость работ, чел/дн.

Для того чтобы реализовать весь план работ, перечисленных в таблице 2, необходимы следующие специалисты: исполнитель и научный руководитель. Сделаем расчет длительности этапов в рабочих днях, далее переведем в календарные дни по следующей формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} * K_{Д}, \quad (3)$$

где $t_{ож}$ – трудоемкость работы, чел/дн;

$K_{ВН}$ – коэффициент выполнения работ ($K_{ВН} = 1$);

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ($K_{Д} = 1.2$).

Расчет продолжительности в календарных днях рассчитывается по следующей формуле:

$$T_{КД} = T_{РД} * T_{К}, \quad (4)$$

где $T_{РД}$ – продолжительности выполнения этапа в рабочих днях;

$T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле (5):

$$T_{К} = \frac{365}{365-52-14} = 1,22 \quad (5)$$

Таблица 4.7 – Трудозатраты на выполнение проекта

Этапы	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
Постановка цели и задач	НР	2	4	2,8	3,4	0	6	0
Подбор и изучение теоретического материала по тематике	НР, И	10	12	10,8	1,3	11,6	2	15
Разработка календарного плана	НР, И	1	3	1,8	1,8	0,43	3	1

Проектирование схемы интеграции	И	8	10	8,8	0	10,5	0	13
Проектирование базы данных	И	3	5	3,8	0	4,56	0	6
Реализация веб-сервиса	И	30	35	32	0	38,4	0	47
Разработка административной панели	И	20	30	24	0	28,8	0	36
Оформление пояснительной записки	И	4	6	4,8	0	5,76	0	8
Подведение итогов	НР, И	2	3	2,4	1,4	1,44	2	2
Итого:		80	108	91,2	8	101	13	128

Полученные значения этапов трудоемкости по исполнителям Т_{КД} позволяют построить линейный график осуществления данного проекта (рисунок 4.2).

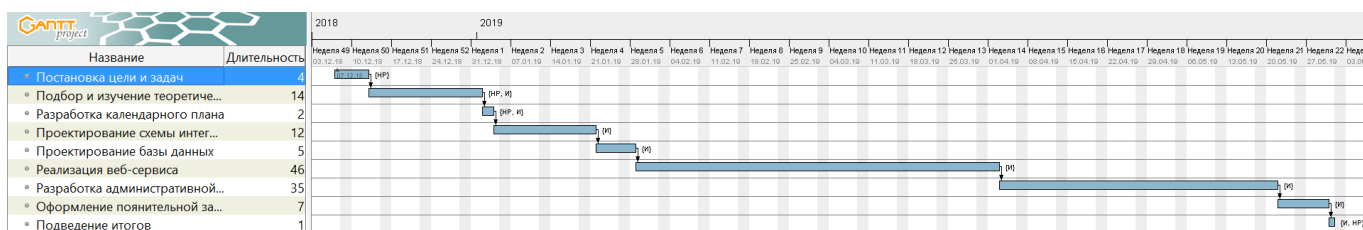


Рисунок 4.2 – График проведения работ

4.3 Расчет сметы затрат на выполнение проекта

Затраты на создание проекта все расходы, необходимые для реализации комплекса работ, входящих в данную разработку. Сметная стоимость ее выполнения рассчитывается по следующим статьям затрат:

1. Материальные затраты
2. Амортизация
3. Затраты на заработную плату
4. Отчисления на социальные нужды
5. Накладные затраты

4.3.1 Расчет материальных затрат

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ.

Таблица 4.8 – Материальные затраты

Наименование	Ед. измерения	Количество	Цена за ед., руб	Затраты на материалы, руб
Бумага формата А4 для принтера	Уп.	1	310	310
Картридж для принтера	Шт.	1	730	730
Итого				1040

Транспортно-заготовительные расходы (ТРЗ) составляют 5% от отпускной цены материалов. Расходы на материалы с учётом ТРЗ:

$$C_{\text{МАТ}} = 1040 * 1,05 = 1092 \text{ руб}$$

4.3.2 Расчет заработной платы

Для выполнения расчета основной заработной платы осуществляется на основе трудоемкости каждого этапа и величины месячного оклада исполнителя. Средняя заработная плата должна рассчитываться по следующей формуле

$$ЗП_{\text{дн-т}} = \frac{МО}{D_{\text{раб}}}, \quad (6)$$

где МО – месячный оклад,

$D_{\text{раб}}$ – количество рабочих дней в месяце.

В таблице 4.9 приведен расчет затрат на полную заработную плату. При данном расчете учитывается, что рабочих дней в году – 299, при 6-дневной рабочей неделе, в месяце всего 24,92 рабочих дня. Затраты на выполнение данной работы по каждому исполнителю берутся из таблицы по трудозатратам.

Чтобы рассчитать в составе заработной платы таких пунктов как премия, дополнительная дополнительная зарплата и районной надбавки, используются

следующие коэффициенты: $K_{\text{ПР}} = 1,1$, коэффициент дополнительной заработной платы $K_{\text{ДОП}} = 1,188$ для шестидневной рабочей недели и районный коэффициент $K_{\text{РК}} = 1,3$. Формула расчета:

$$K = K_{\text{ПР}} * K_{\text{РК}} * K_{\text{ДОП}}; \quad (7)$$

Далее рассчитываем ЗП за проект по формуле

$$\text{ЗП}_{\text{П}} = \text{ЗП}_{\text{ДН-Т}} * K * T \quad (8)$$

где T- затраты времени, дни

Таблица 4.9 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./день	Затраты времени, дни	K	Фонд з/платы, руб.
НР	29568	1186,5	8	1,698	16117,4
И	21760	873,2	101	1,698	149751
Итого:					165868,4

4.3.3 Расчет затрат на электроэнергию

А также в этот вид расходов включается электроэнергия, потраченная на работу используемого для проекта оборудования, расход электроэнергии на освещение не учитываются.

Расчет производится по формуле:

$$\text{Э}_{\text{об}} = P_{\text{об}} * t_{\text{об}} * \text{Ц}_{\text{э}}, \quad (9)$$

где $P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$t_{\text{об}}$ – время работы оборудования, час;

$\text{Ц}_{\text{э}}$ – тариф на 1 кВт*час.

Время работы оборудования рассчитывается исходя из итоговых данных таблицы из расчета, что продолжительность рабочего дня составляет 8 часов, формула:

$$t_{\text{об}} = T_{\text{РД}} * K_t, \quad (10)$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{\text{РД}}$, в данном случае будет приравниваться к 0,7.

Мощность определяется по формуле:

$$P_{\text{ОБ}} = P_{\text{НОМ.}} * K_C, \quad (11)$$

где $P_{\text{НОМ.}}$ – номинальная мощность оборудования, кВт;

$K_C \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_C = 1$.

Для $C_3 = 5,257$ руб./кВт*час (с НДС) заполним таблицу 4.10.

Таблица 4.10 – Затраты на электроэнергию.

Наименование оборудования	Время работы оборудования, $t_{\text{об}} = T_{\text{рД}} * K_t$, час	Потребляемая мощность $P_{\text{об}}$, кВт	Затраты $\Delta_{\text{об}}$, руб.
ПК	565,6	0,3	892

4.3.4 Расчет затрат на социальные нужды

Данная статья расходов отвечает за отчисления на социальные нужды который включает в себя отчисления в пенсионный фонд, на социальное и медицинское страхование.

Отчисления рассчитываются по следующей формуле (12):

$$C_{\text{соц}} = K_{\text{соц}} * C_{\text{осн}}; \quad (12)$$

где $K_{\text{соц}}$ – коэффициент, учитывающий размер отчислений из заработной платы, он равен 30%.

Далее вычислим социальные отчисления для полученных данных:

$$C_{\text{соц}} = 0,3 * 165868,4 = 49760,4 \text{ руб}$$

4.3.5 Расчет амортизационных расходов

Далее проведем расчет амортизации используемого оборудования за время выполнения проекта. Для этого используется следующая формула:

$$C_{\text{ам}} = \frac{C_{\text{об}} * t_{\text{рф}}}{299}, \quad (13)$$

где $C_{\text{ОБ}}$ – балансовая стоимость единицы оборудования с учетом ТЗР, стоимость ПК инженера – 52000 руб;

$t_{\text{рф}}$ – фактическое время работы оборудования в ходе выполнения проекта;

Для расчета амортизации ПК время выполнения проекта необходимо рассчитать амортизацию за год для данного оборудования

$$C_{\text{АМ ПК}} = \frac{52000}{3} = 17333 \text{ р.}$$

Таким образом, находим амортизационные расходы во время выполнения проекта:

$$C_{\text{АМ}} = \frac{17333 * 6}{12} = 8666,5 \text{ р.}$$

4.3.6 Расчет накладных расходов

Накладные расходы следует принять равными 16% от суммы всех предыдущих расходов. Рассчитаем расходы на выполнение тех пунктов проекта, которые не были учтены ранее:

$$C_{\text{НР}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}} + C_{\text{св}} + C_{\text{нп}}) * 0,16 \quad (14)$$

Для данного проекта:

$$C_{\text{НР}} = (1092 + 165868,4 + 49760,4 + 892 + 8666,5) * 0,16 = 36204,6 \text{ руб}$$

4.3.7 Расчет планируемой прибыли

Если рентабельность установим 20% от расходов на разработку:

$$262483,9 * 0,2 = 52496,7 \text{ руб.}$$

4.3.8 Расчет НДС

НДС составляет 20% от суммы затрат на разработку, а также полученной прибыли. Таким образом, НДС составляет

$$(262483,9 + 52496,7) * 0,2 = 62996,1 \text{ руб.}$$

4.3.9 Цена разработки

Данный показатель равен сумме полной себестоимости, прибыли и НДС. Следовательно

$$C_{\text{НИР(КР)}} = 262483,9 + 52496,7 + 62996,1 = 377976,7 \text{ руб.}$$

4.3.10 Расчет общей себестоимости разработки

Определим общую себестоимость, объединив все затраты в таблицу 4.11. Затраты по отдельным статьям для текущего проекта рассчитывались ранее.

Таблица 4.11 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материальные расходы	$C_{\text{мат}}$	1092
Основная заработная плата	$C_{\text{зп}}$	165868,4
Расходы на электроэнергию	$\text{Э}_{\text{ОБ}}$	892
Отчисления в социальные фонды	$C_{\text{соц}}$	49760,4
Амортизационные отчисления	$C_{\text{ам}}$	8666,5
Накладные расходы	$C_{\text{нр}}$	36204,6
Итого:		262483,9
Прибыль	P	52496,7
НДС	НДС	62996,1
Итого:		377976,7

Таким образом, стоимость разработки проекта составили $C = 377976,7$ рублей. Рассчитанный бюджет не превышает бюджета в 400000 рублей, указанного в выдержках из устава проекта, представленного в таблице 4.5.

4.4 Потенциальные риски

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать последствия, которые повлекут за собой нежелательные эффекты.

Риски, которые могут возникнуть при реализации проекта, приведены в таблице 4.12.

Таблица 4.12 – Риски проекта

Риск	Потенциальное воздействие	Вероятность наступления	Влияние риска	Уровень риска	Условия наступления	Мероприятия по устранению
Отсутствие финансирования	Снижение качества и увеличение сроков выхода на рынок	средняя	среднее	высокий	Недостаточно проработанная стратегия привлечения средств	Привлечение потенциальных инвесторов
Низкий темп разработки	Увеличение сроков разработки	средняя	высокое	средний	Недостаточная квалификация исполнителей	Найм квалифицированных специалистов
Недостаток кадров	Снижение производительности и качества	низкая	высокое	высокий	Отсутствие финансирования	Увеличение штата разработчиков
Ошибки при разработке	Снижение качества разработки	средняя	высокое	средний	Отсутствие или недостаточное количество проведенных тестирований	Увеличение команды тестировщиков
Потеря актуальности	Увеличение сроков разработки	низкая	высокое	средний	Появление конкурирующих разработок на рынке	Улучшение системы за счет новых уникальных подсистем

4.5 Определение ресурсной и финансовой эффективности проекта

Для определения эффективности необходимо было рассчитать интегральный показатель. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

Формула интегрального финансового показателя разработки:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}} \quad (15)$$

Где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;
 Φ_{pi} – стоимость i -го варианта исполнения;
 Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Величина интегрального финансового показателя разработки отражает соответствующие численное увеличение бюджета затрат на разработку в разгах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разгах (значение меньше единицы, но больше нуля).

Таким образом, интегральный финансовый показатель данной разработки равен:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{377976,7}{400000} = 0,94$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_m^a = \sum_{i=1}^n a_i * b_i^a, \quad (16)$$

Где I_m – интегральный показатель ресурсоэффетивности для исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a – бальная оценка i -го варианта исполнения разработки;

n – число параметров сравнения.

Таблица 4.13 – Сравнительная оценка характеристик исполнения проекта

Критерий	Весовой коэффициент	Текущий проект
Сложность структуры	0.10	4
Функциональная мощность	0.20	5
Кроссплатформенность	0.10	4
Наличие дополнительных подсистем (условий)	0.15	5
Потребность в ресурсах памяти	0.10	2
Конкурентоспособность продукта	0.20	3
Законченность работы	0.10	4
Финансирование	0.05	1

В итоге: $I_{\text{тп}} = 3,8$

Сравнение значений интегральных показателей эффективности позволяет понять, какой вариант решения является максимально эффективным с позиции финансовой и ресурсной эффективности. Но данная задача не имеет особо различающихся по бюджету вариантов.

4.6 Выводы по разделу

В ходе выполнения предпроектного анализа было выявлено, что данное исследование относится к сегменту разработки программного обеспечения, потребителем которого являются коммерческие организации, разрабатывающие различное программное обеспечение и использующие такую систему управления задач, как YouTrack.

Оценка готовности проекта к коммерциализации показала, что перспективность разработки представленного проекта выше среднего, а, следовательно, исследование необходимо продолжать.

SWOT-анализ показал, что ключевой проблемой проекта является отсутствие финансирования, из-за чего затруднительно активно внедрять и расширять данную систему. Рассмотрев слабые стороны проекта и возможности, можно сделать вывод, что необходима стратегия поиска дополнительного финансирования проекта, которое позволит улучшать технологии разработанной системы, что положительно скажется на автоматизации бизнес-процессов предприятия, где система будет интегрирована.

В ходе расчета продолжительности выполнения проекта было выявлено, что на выполнение всех работ уйдет 128 календарных дней.

В ходе расчета сметы затрат на выполнение проекта была выявлена цена разработки проекта, которая равна 377976,7 руб.

Вероятность наступления рисков ниже среднего, однако, данные риски имеют высокое влияние на проект.

Таким образом, было выявлено, что разработка данного проекта является достаточно перспективной.

5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Для рассмотрения производственной безопасности проекта необходимо выявить вредные и опасные факторы, которые могут возникнуть на рабочем месте, и описать мероприятия по защите исследователя и пользователей конечных продуктов от действия опасных и вредных факторов.

Данный программный продукт планируется использовать на предприятии ОАО «ТомскНИПИнефть», которое географически располагается в городе Томск Томской области. Модуль интеграции предназначен для передачи данных из вики-системы XWiki в систему управления задачами YouTrack. Основным потребителем данной продукции будет являться предприятие ОАО «ТомскНИПИнефть».

5.1 Правовые и организационные вопросы обеспечения безопасности

Правовую основу охраны труда составляют законодательные акты и нормативные правовые документы по охране труда, имеющие иерархическую структуру, обладающие различной юридической силой. Нормы трудового права регулируют любые отношения, связанные с использованием личного труда.

Основной задачей трудового договора является создание необходимых организационно-правовых условий для достижения оптимального согласования интересов сторон трудовых отношений.

По заключенному, на основании Трудового кодекса РФ от 30.12.2001 N 197-ФЗ, трудовому договору работник обязан:

- лично выполнять определенную трудовым договором функцию с соблюдением внутреннего трудового распорядка, установленного Работодателем;
- добросовестно исполнять свои трудовые обязанности, приказы, распоряжения и указания своего непосредственного руководителя;

- бережно относиться к имуществу Работодателя, в том числе находящемуся в его пользовании, правильно и по назначению использовать переданные ему для работы оборудование, приборы;

- нести иные обязанности, предусмотренные действующим законодательством Российской Федерации и локальными нормативными актами Работодателя.

В свою очередь работодатель обязан:

- предоставить работнику работу по обусловленной трудовой функции,

- обеспечить условия труда, предусмотренные трудовым законодательством и иными правовыми нормативными актами, соглашениями, содержащими нормы трудового права, коллективным договором;

- нести иные обязанности, предусмотренные действующим законодательством Российской Федерации и локальными нормативными актами Работодателя.

Также, работник имеет право:

- изменять и расторгать трудовой договор в порядке и на условиях, определенных трудовым законодательством Российской Федерации,

- своевременно и в полном объеме получать заработную плату в соответствии со своей квалификацией, сложностью труда, количеством и качеством выполненной работы,

Работодатель имеет право:

- изменять и расторгать договор в порядке и на условиях, определенных трудовым законодательством Российской Федерации,

- требовать от Работника и контролировать исполнение им трудовых обязанностей и бережного отношения к имуществу нормативных актов, трудовой дисциплины, правил техники безопасности, производственной санитарии и противопожарной защиты и иных локальных нормативных актов Общества;

– привлекать Работника к дисциплинарной и материальной ответственности за неисполнение или ненадлежащее исполнение Работником трудовых обязанностей в порядке, установленным трудовым законодательством Российской Федерации.

На основании ТК РФ статьи 135, трудовым договором устанавливается заработная плата в соответствии с действующими у данного работодателя системами оплаты труда. Заработная плата выплачивается Работнику два раза в месяц в порядке и сроки, установленные Правилами внутреннего трудового распорядка в соответствии с действующим законодательством: за первую половину месяца – 30 числа текущего месяца, окончательный расчет – 15 числа месяца следующего за отчетным.

В соответствии с ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ)» к рабочему месту предъявляются следующие основные эргономические требования:

– Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы;

– Конструкцией рабочего места должно быть обеспечено выполнение трудовых операций в пределах зоны досягаемости моторного поля;

– Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы;

– Рабочее место для выполнения работ сидя организуют при легкой работе, не требующей свободного передвижения работающего, а также при работе средней тяжести в случаях, обусловленных особенностями технологического процесса;

– При проектировании оборудования и организации рабочего места следует учитывать антропометрические показатели женщин (если работают

только женщины) и мужчин (если работают только мужчины); если оборудование обслуживают женщины и мужчины - общие средние показатели женщин и мужчин;

– Конструкцией рабочего места должно быть обеспечено оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, сиденья и пространства для ног.

5.2 Производственная безопасность

Выявление на рабочем месте факторов производственной среды и трудового процесса, источников вредных и (или) опасных факторов осуществляется путем изучения представляемых работодателем:

- технической (эксплуатационной) документации на производственное оборудование (машины, механизмы, инструменты и приспособления), используемое работником на рабочем месте;
- должностной инструкции и иных документов, регламентирующих обязанности работника;
- характеристик применяемых в производстве материалов и сырья;
- результатов ранее проводившихся на данном рабочем месте исследований (испытаний) и измерений вредных и (или) опасных факторов.

Основные вредные и опасные факторы при выполнении технологических процессов разработки программного обеспечения представлены в таблице 5.1.

Таблица 5.1 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разраб отка	Изгото вление	Эксплу атация	
1.Отклонение показателей микроклимата	-	-	-	СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений.

2. Превышение уровня шума	-	-	-	ГОСТ 12.1.029-80 Средства и методы защиты от шума.
3.Отсутствие или недостаток естественного света	+	-	+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95
4.Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека	-	-	-	ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов

5.2.1 Отклонение показателей микроклимата

Микроклимат помещения – это комплекс метеорологических условий в данном помещении. Компьютеры могут привести к увеличению температуры и снижению относительной влажности в помещении. В СанПиН 2.2.4.548 – 96 установлены величины параметров микроклимата, создающие комфортные условия [17].

Требования к микроклимату определяются исходя из категории тяжести работ. Работа разработчика-программиста относится к легкой категории 1Б [17], оптимальные и допустимые значения микроклимата представлены в таблице 5.2.

Таблица 5.2 - Оптимальные и допустимые значения микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	23-25	20-24	40-60	0,1
Тёплый	20-22	21-25	40	0,1

Если температура воздуха отличается от нормальной, то время пребывания в таком помещении должно быть ограничено в зависимости от категории тяжести работ. Температура в рассматриваемом помещении в холодное время года может опускаться до 19-21 °С, а в теплое время года подниматься до 25-28 °С. Данные показатели соответствуют допустимым

значениям температуры, рекомендуемое время работы при температуре воздуха ниже допустимых величин представлено в таблице 5.3, при температуре выше допустимых величин представлено в таблице 5.4.

Таблица 5.3 – Рекомендуемое время работы при температуре воздуха ниже допустимых величин

Температура воздуха, °С	Время пребывания, не более, ч
17,0	6
18,0	7

Таблица 5.4 – Рекомендуемое время работы при температуре воздуха выше допустимых величин

Температура воздуха, °С	Время пребывания, не более, ч
30,0	5
29,5	5,5
29,0	6

К мероприятиям по оздоровлению воздушной среды в производственном помещении относятся правильная организация вентиляции и кондиционирования воздуха, отопление помещений. В рассматриваемом помещении вентиляция осуществляется естественным и механическим путём. В зимнее время в помещении предусматривается система отопления. Это обеспечивает нормальное состояние здоровья работников в помещении.

5.2.2 Превышение уровня шума

Для исследуемого объекта (веб-сервиса интеграции данных) основными источниками шумов, а также и вибрации являются производственные процессы, сопровождающиеся шумом, такая как, работа с компьютером. Говоря, о действии шума на организм, следует иметь в виду, что он оказывает как местное, так и общее воздействие. Неблагоприятно отражается шум на нервной системе, вызывая головные боли, бессонницу, ослабление внимания,

замедление психических реакций, пониженный аппетит, боли в ушах, что, в конечном счете, приводит к понижению работоспособности [18].

Допустимые уровни звука и звукового давления для рабочего места разработчика-программиста согласно ГОСТ 12.1.003-83 ССБТ «Общие требования безопасности» представлены в таблице 5.5 [19].

Таблица 5.5 – Предельно допустимые уровни звука

Вид трудовой деятельности/ Частоты	Уровни звука и звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц								
	31,5	63	125	250	500	1000	2000	4000	8000
Научная деятельность, проектирование, программирование, Рабочие места проектно-конструкторских бюро, программистов вычислительных машин и т.д.	86	71	61	54	49	45	42	40	38

Допустимый уровень звукового давления колеблется от 38 дБ до 86 дБ при частоте от 8000 Гц до 31,5 Гц, соответственно.

Для снижения уровня шума, производимого персональными компьютерами, рекомендуется регулярно проводить их техническое обслуживание: чистка от пыли, замена смазывающих веществ; также применяются звукопоглощающие материалы [20].

5.2.3 Отсутствие или недостаток естественного света

Недостаточная освещенность рабочей зоны является вредным производственным фактором, который может привести к быстрому утомлению и снижению работоспособности, обеспечение обеспечения нормативных условий работы в помещениях и проводится в соответствии с СанПиН 2.2.1/2.1.1.1278-03.

Свет влияет на физиологическое состояние человека, правильно организованное освещение стимулирует протекание процессов высшей

нервной деятельности и повышает работоспособность. При недостаточном освещении человек работает менее продуктивно, быстро устает, растет вероятность ошибочных действий, что может привести к травматизму. Основным документом, регламентирующим нормы освещенности, является СНиП 23-05-95* [21].

Естественное и искусственное освещение в помещениях регламентируется нормами в зависимости от характера зрительной системы и вида освещения, фона, контраста объекта с фоном. Помещения с ПК должны иметь естественное и искусственное освещение. Естественное освещение должно осуществляться через световые проёмы, ориентированные преимущественно на север и северо-восток. В данном производственном помещении используется искусственное и естественное освещение. Естественное освещение – боковое, осуществляемое через оконные проемы в боковых наружных стенах. Искусственное освещение – общее равномерное (световой поток равномерно распределяется по помещению без учета расположения оборудования),

Основным показателем качества освещения является освещенность E - поверхностная плотность светового потока. По характеристике зрительной работы труд программиста относится к разряду III подразряду Г (высокой точности), т.е. наименьший размер объекта различения от 0,3 до 0,5 мм (точка) [22]. Это значит, что нормативное значение освещенности рабочего места должно быть 200 лк [21].

При эксплуатации установок искусственного освещения необходимо регулярно производить очистку светильников от загрязнений, своевременную замену перегоревших или отработавших свой срок службы ламп, контроль напряжений в осветительной сети, регулярную окраску или побелку стен и потолка. Периодически, но не реже одного раза в год, должен проводиться контроль освещенности на рабочих поверхностях с помощью фотоэлектрических люксометров Ю-116, Ю-117 и др. Для обеспечения

нормативной освещенности необходимо использовать совмещенное освещение, при котором естественное дополняется искусственным освещением.

5.2.4 Электромагнитное излучение

Электромагнитное излучение - распространяющееся в пространстве возмущение электрических и магнитных полей [23]. Источниками электромагнитного излучения в данном исследовании являются мониторы и системный блок.

Оценка величины уровней ЭМП, проведенная по паспортным данным компьютера и монитора, показала их соответствие нормам ТСО-03 и СанПиН 2.2.2/2.4.1340-03 [22]. В таблице 5.6 приведены нормы уровня ЭМП, которым соответствует техника в кабинете.

Таблица 5.6 – Допустимые уровни ЭМП, создаваемых ПК

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Для того, чтобы снизить воздействие таких видов излучения, рекомендуют применять такие мониторы, у которых уровень излучения понижен (MPR-II, ТСО-92, ТСО-99), а также установить защитные экраны и соблюдать режимы труда и отдыха.

5.3 Экологическая безопасность

Экологическая безопасность и охрана окружающей среды являются одними из важнейших факторов при выполнении работ любого характера. При работе в офисном помещении за персональным ПК отсутствуют выбросы в окружающую среду и влияния на жилищную зону.

Поскольку при разработке данной магистерской диссертации использовался компьютер, необходимо помнить о правильной утилизации компьютерного лома после выхода из строя данного ПК. В соответствии с постановлением правительства №340 СанПиН 2.2.2. 542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы» юридическим лицам запрещено самостоятельно утилизировать компьютерную технику. Необходимо найти организацию, которая занимается утилизацией в частном порядке. Это относится к следующим видам отходов:

- Твердые отходы, относящихся к IV классу опасности (системный блок компьютера, принтеры, сканеры, клавиатура, манипулятор «мышь») [24];
- Жидкие отходы: сточные воды;
- Люминесцентные лампы [25].

5.4 Безопасность в чрезвычайных ситуациях

Одними из наиболее вероятных и разрушительных видов чрезвычайных ситуаций являются пожар или взрыв на рабочем месте.

Всякий работник при обнаружении пожара должен [26]:

- незамедлительно сообщить об этом в пожарную охрану;
- принять меры по эвакуации людей, каких-либо материальных ценностей согласно плану эвакуации;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

При возникновении пожара должна сработать система пожаротушения, передав на пункт пожарной станции сигнал о ЧС. В случае если система не сработала, то необходимо самостоятельно произвести вызов пожарной службы по телефону 101, сообщить точный адрес места возникновения ЧС и ожидать приезда специалистов.

Рабочее место располагается по адресу ул. Мокрушина 9, стр. 16 (Бизнес-центр «Вертикаль») в 332 кабинете. На рисунке 5.1 представлен план эвакуации правого крыла третьего этажа бизнес-центра «Вертикаль».

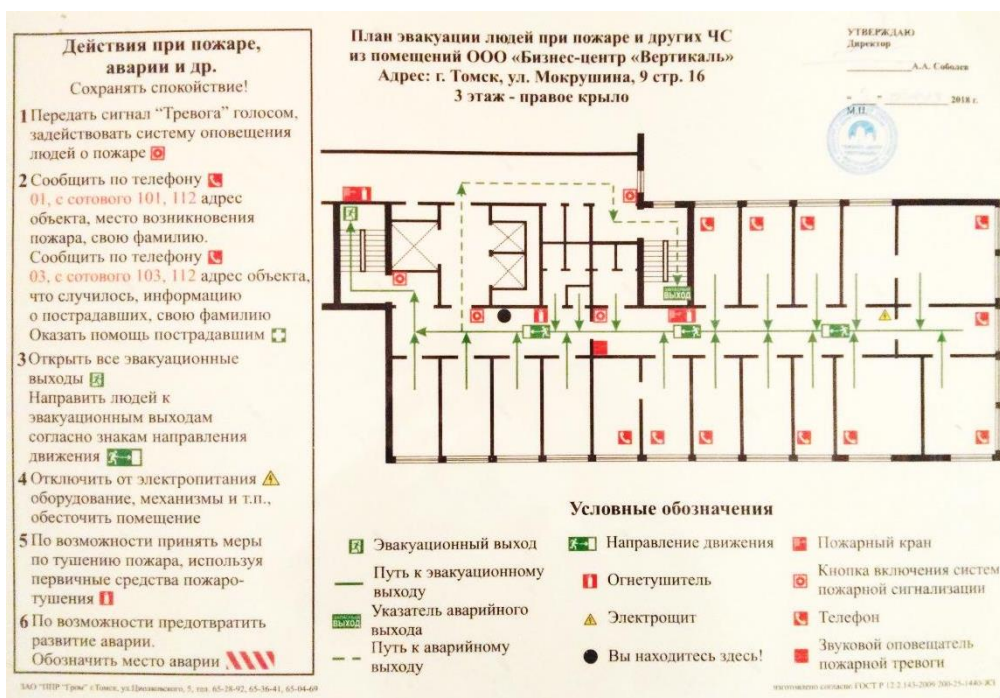


Рисунок 5.1 – План эвакуации при чрезвычайных ситуациях

ЗАКЛЮЧЕНИЕ

Практическим результатом моей работы является модуль интеграции для взаимодействия информационных систем YouTrack и XWiki. Он включает в себя веб-сервис, административную панель и базу данных.

В процессе выполнения данной работы, были получены следующие результаты:

1. Проанализированы основные проблемы при интеграции систем.
2. На стадии проектирования модуля интеграции была изучена структура системы трекинга задач YouTrack.
3. Спроектированы IDEF0-диаграммы, отражающие процесс взаимодействия между модулем интеграции и системой YouTrack.
4. Спроектирована схема взаимодействия двух систем и реляционная модель базы данных веб-сервиса.
5. Разработаны спецификации требований к модулю интеграции.
6. Для программной реализации веб-сервиса была выбрана среда разработки PhpStorm. В результате был создан веб-сервис на PHP-фреймворке Yii2 по шаблону проектирования MVC.

Подробнее о взаимодействии XWiki и разработанного модуля интеграции представлено в работе магистранта гр. 8ИМ71 Дергачева А.О.

Функциональность модуля интеграции может быть расширена путем добавления веб-сервису новых API методов и возможностей по настройке соединения с системой трекинга задач YouTrack.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Добровольский А. Интеграция приложений: методы взаимодействия, топология, инструменты / [Электронный ресурс] Открытые системы. СУБД | Издательство «Открытые системы». Режим доступа: <https://www.osp.ru/os/2006/09/3776464/> (дата обращения 11.03.2019).
2. Кручинин С.В. Системы управления задачами юзабилити и значимость для современного бизнеса / [Электронный ресурс] КиберЛенинка. Режим доступа: <https://cyberleninka.ru/article/n/sistemy-upravleniya-zadachami-yuzabiliti-i-znachimost-dlya-sovremennogo-biznesa> (дата обращения 11.03.2019).
3. «Проблемы интеграции» // [Электронный ресурс] Режим доступа: <http://ict.informika.ru/vconf/files/10137.pdf> (дата обращения 17.02.2019).
4. «Методы и подходы к интеграции систем» // [Электронный ресурс] Режим доступа: https://flexberry.github.io/ru/gbt_integration-methods.html#4-интеграция-бизнес-процессов (дата обращения 21.05.2019).
5. YouTrack / [Электронный ресурс] Starpack – Поиск и подбор сервисов для бизнеса. Режим доступа: <https://startpack.ru/application/youtrack> (дата обращения 11.03.2019).
6. Инструменты для совместной работы YouTrack / [Электронный ресурс] JetBrains — Ведущий мировой производитель профессиональных средств разработки. Режим доступа: <https://jetbrains.ru/products/youtrack/> (дата обращения 11.03.2019).
7. Паттерны для новичков: MVC vs MVP vs MVVM / [Электронный ресурс] Хабр — крупнейший в Европе ресурс для IT-специалистов. Режим доступа: <https://habr.com/ru/post/215605/> (дата обращения 21.05.2019).
8. «Архитектура REST» / [Электронный ресурс] Хабр — крупнейший в Европе ресурс для IT-специалистов. Режим доступа: <https://habr.com/ru/post/38730/> (дата обращения 21.05.2019).
9. REST vs SOAP. Часть 1. Почувствуйте разницу / [Электронный ресурс] Хабр — крупнейший в Европе ресурс для IT-специалистов. Режим доступа: <https://habr.com/ru/post/131343/> (дата обращения 21.05.2019).

10. PHP vs Python vs Ruby для разработки веб-приложений: подробное сравнение / [Электронный ресурс] Режим доступа: <https://8d9.ru/php-vs-python-vs-ruby-dlya-razrabotki-veb-prilozhenij-podrobnoe-sravnenie> (дата обращения 21.05.2019).
11. Преимущества и недостатки использования php-фреймворков // [Электронный ресурс] Режим доступа: <https://www.reclamare.ua/blog/preimushhestva-i-nedostatki-ispolzovaniya-php-frejmvorkov-v-processe-razrabotki-sajtov/> (Дата обращения 19.05.2019).
12. «Использование систем управление сайтом и фреймворков при разработке сайта» // [Электронный ресурс] Режим доступа: https://pugofka.com/blog/technology/ispolzovanie_cms_and_framework_pri_razrabotke_sayta/ (дата обращения 25.04.2019).
13. «Как выбрать PHP-фреймворк» // [Электронный ресурс] Режим доступа: <https://habrahabr.ru/company/nixsolutions/blog/329718/> (дата обращения 25.04.2019).
14. Модель-Представление-Контроллер (MVC) / [Электронный ресурс] Режим доступа: <https://www.yiiframework.com/doc/guide/1.1/ru/basics.mvc> (дата обращения 21.05.2019).
15. Протасевич С. Guzzle – простой PHP HTTP Клиент (часть 1) / [Электронный ресурс] Видеоуроки по созданию сайта для бизнеса. Бесплатные видеоуроки. Режим доступа: <https://ruseller.com/lessons.php?rub=37&id=1955> (дата обращения 11.03.2019).
16. Программирование с Yii2: Маршрутизация и создание URL-адреса / [Электронный ресурс] Режим доступа: <https://code.tutsplus.com/ru/tutorials/programming-with-yii2-routing-and-url-creation--cms-26869> (дата обращения 21.05.2019).
17. СанПиН 2.2.4.548 – 96. Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрава России, 1997. – 20 с.

18. Борьба с шумом на производстве: Справочник / Е.Я. Юдин, Л.А. Борисов; Под общ. ред. Е.Я. Юдина – М.: Машиностроение, 1985. – 400с.
19. ГОСТ 12.1.003-2014. ССБТ. Шум. Общие требования безопасности.
20. СНиП 23-03-2003. Защита от шума.
21. СНиП 23-05-95*. Естественное и искусственное освещение. – М.: Центр проектной продукции в строительстве, 2011. – 70 с.
22. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрава России, 2003. – 54 с.
23. Безопасность жизнедеятельности. /Под ред. Н.А. Белова - М.: Знание, 2000 - 364с.
24. СанПин 2.17.1322-03. Гигиенические требования к размещению и обезвреживанию отходов производства и потребления: Санитарно-эпидемиологические правила и нормативы. – М.: Федеральный центр госсанэпиднадзора Минздрава России, 2004. – 16 с.
25. ГОСТ Р 51768-2001 Ресурсосбережение. Обращение с отходами. Методика определения ртути в ртутьсодержащих отходах. Общие требования.
26. Постановление Правительства РФ от 25 апреля 2012 года N 390 «О противопожарном режиме» (с изменениями на 7 марта 2019 года).

ПРИЛОЖЕНИЕ А

(Обязательное)

Раздел II

DESIGNING INTEGRATION MODULE

Студент

Группа	ФИО	Подпись	Дата
8ИМ71	Евстафьев Сергей Николаевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Евгений Александрович	к.т.н.		

Консультант отделения ИТ

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Евгений Александрович	к.т.н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ	Сидоренко Татьяна Валерьевна	к.п.н.		

2.1 Structure of the YouTrack task tracking system

YouTrack is a commercial error tracking system, task and project management software developed by JetBrains. YouTrack supports search queries, autocompletion, manipulation with task sets, setting up a set of task attributes, creating custom workflows and implements an approach based on preferential use of the keyboard. The program interface is shown in Figure 2.1.

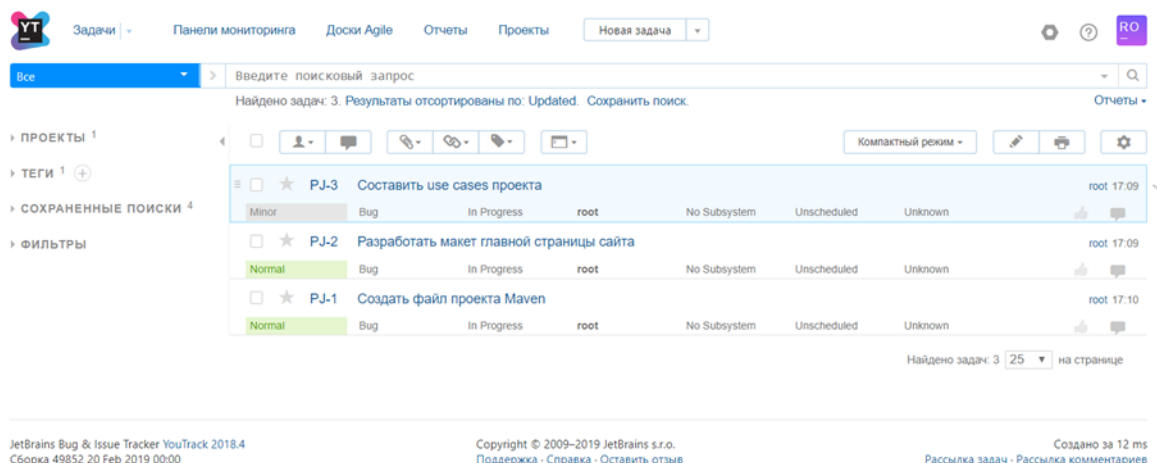


Figure 2.1 – System Interface

YouTrack is designed in accordance with the language-oriented programming paradigm, uses JetBrains MPS and a set of its own domain-oriented programming languages developed by JetBrains. The system uses the built-in JetBrains Database. It is a key-value transactional repository. It uses the REST style for remote procedure calls.

YouTrack integrates with version control systems such as Subversion, CVS, Rational ClearCase, Visual SourceSafe, Mercurial, Git and Perforce using TeamCity. There is also the ability to integrate with the GitHub service. It is possible to integrate with TeamCity, IntelliJ IDEA, TestLink, TestRail, support for user authentication using Google Account, Yahoo !, OpenID and LDAP. YouTrack can import tasks created in other bug tracking systems: JIRA, FogBugz, Bugzilla, MantisBT, Trac, CSV files and Google Code bug tracker. YouTrack provides a library written in Python for importing tasks created in other error tracking systems [5].

The system setup process is very simple in the version of YouTrack inCloud. You only need to register on the official website and have a permanent Internet connection. All data will be stored in the cloud with daily data backup.

YouTrack is distributed under a commercial license in five different ways, differing in the number of supported users who can work with the system. JetBrains provides YouTrack for free use by developers of open source projects and for training [6].

The web service works with two standard entities provided by the YouTrack structure: tasks and projects. Table 2.1 presents the attributes of the Project entity.

Table 2.1 – Project Entity Attributes

Attribute	Type	Description
startingNumber	Numeric	Starting ID number when creating new tasks
shortName	String	Project ID
description	String	Project Description
leader	User object	Project Manager
createdBy	User object	Project Creator
issues	Array	Project Task List
customFields	Array	Custom Attribute List
archived	Logical	Stay project in the archive
template	Logical	The project is a template.
iconUrl	String	Path to the project icon

Table 2.2 presents the attributes of the Issue entity.

Table 2.2 – Issue Entity Attributes

Attribute	Type	Description
idReadable	String	Task ID
created	Numeric	Creation date
updated	Numeric	Last updated task date
resolved	Numeric	Date of problem solving
numberInProject	Numeric	Task sequence number
project	Project object	The project, in which the task was created
summary	String	Task title
description	String	Task Description
reporter	User object	Task creator
updater	User object	Last user to update task

visibility	Visibiliy object	Task visibility settings for different user groups
votes	Numeric	The number of votes for the task
comments	Array	List of comments for the task
commentsCount	Numeric	Number of comments
tags	Array	List of tags to the task
links	Array	List of links associated with the task
customFields	Array	Custom Attribute List
voters	IssueVoters object	Information about voting users
watchers	IssueWatchers object	Information about users who track the task
attachments	Array	List of materials for the task
subtasks	IssueLink object	List of subtasks for this task
parent	IssueLink object	Parent task of this task

There are over 70 different entities in the YouTrack system, representing objects with their own attributes. REST API YouTrack allows you to work with most of them, but we will consider only work with Project and Issue entities to solve problems in the future, because they are common objects when integrating the two systems.

2.2 Designing the scheme of interaction of systems

The XWiki module is connected via the XWiki API and the extension manager. The XWiki extension interacts with the integration module via the extension API and web service API. Data is transmitted by the XWiki extension in JSON format to a web service, where it is analyzed and transmitted to YouTrack via the REST API.

The general integration scheme is presented in Figure 2.2.

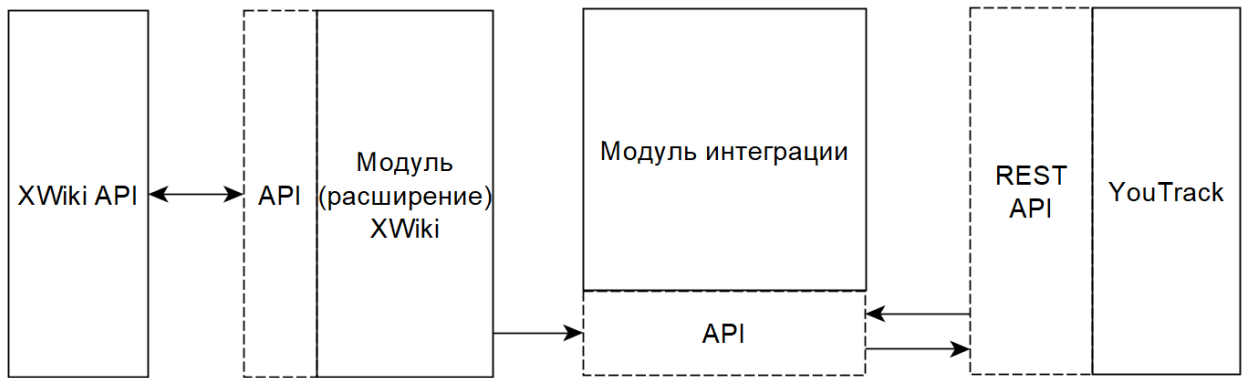


Figure 2.2 – XWiki and YouTrack Integration Chart

The web service adds the YouTrack task ID and the name of the XWiki task from which they came to the relationship table, when creating a new task, before sending data to YouTrack.

The web service analyzes incoming data from the XWiki page addresses, compares according to the correlation table, when updating data, and then updates the information about a specific task by its ID listed in the table.

Figure 2.3 shows the detailed interaction between the integration module and the YouTrack system.

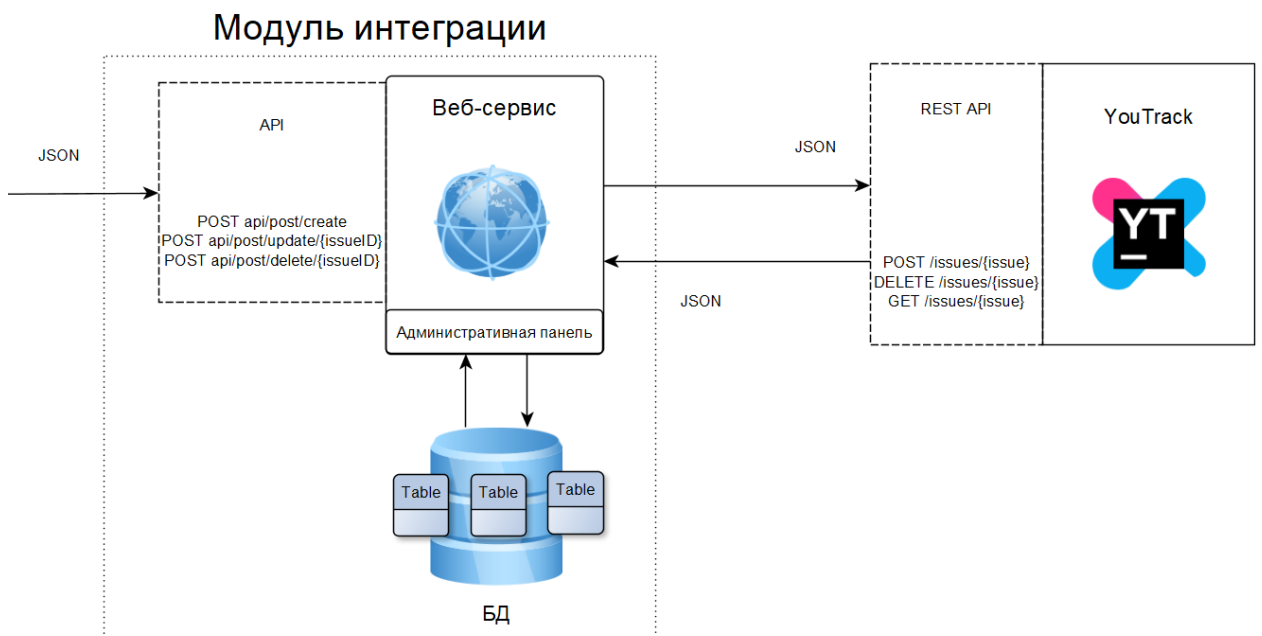


Figure 2.3 – Detailed scheme of interaction between the integration module and YouTrack

The diagram shows some of the developed methods that allow you to use YouTrack REST APIs to transfer data. The data come in JSON format in the web service from the XWiki extension. Data exchange with the YouTrack system is also carried out in JSON format.

2.3 Building a functional diagram with decomposition

The developed system is designed to record, analyze and transfer incoming data to the YouTrack task tracking system.

The system has the following functions:

- 1) Accounting information requirements;
- 2) Accounting information about users;
- 3) Accounting information about events;
- 4) Data comparison
- 5) Forming a request to the YouTrack system.

The web service works in automatic mode, all transferred data is analyzed and transmitted to the YouTrack system. This process will significantly reduce the time to submit low-level requirements to YouTrack and to update this data directly from the XWiki wiki.

The data integration process was modeled using the functional modeling notation IDEF0 based on the functions of the system.

The IDEF0 diagram is shown in Figure 2.4.

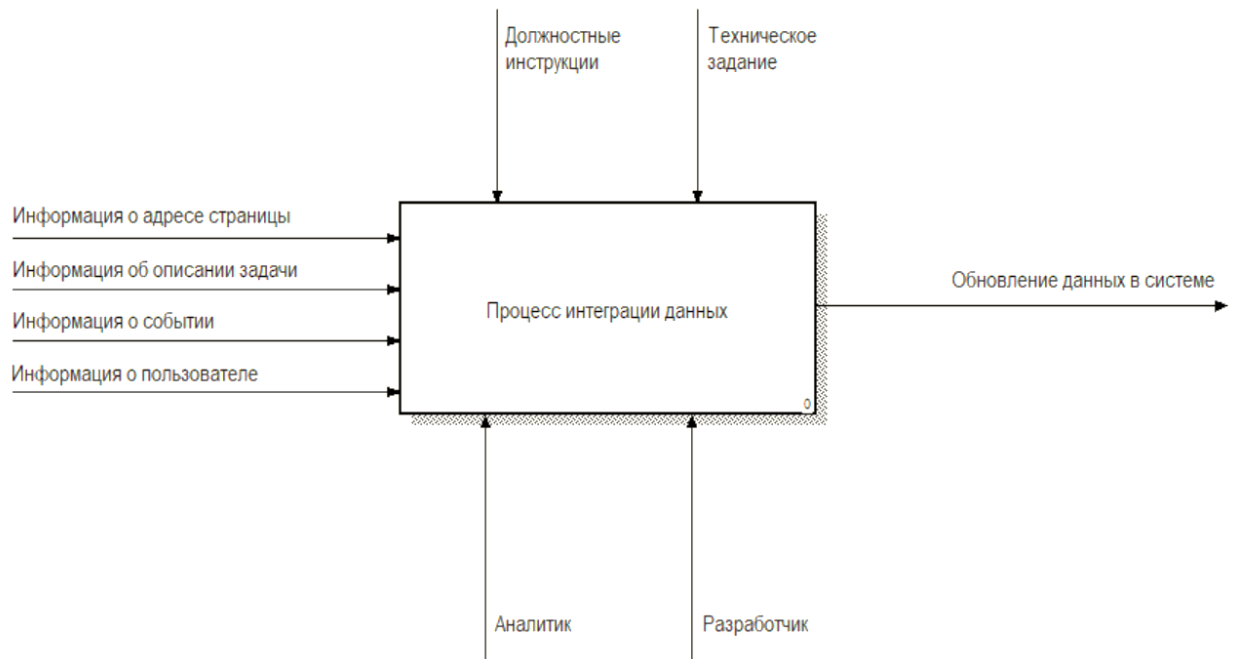


Figure 2.4 – IDEF0-diagram of IS

The whole process of data integration can be decomposed into subsystems. The decomposition of the data integration process is presented in Figure 2.5.

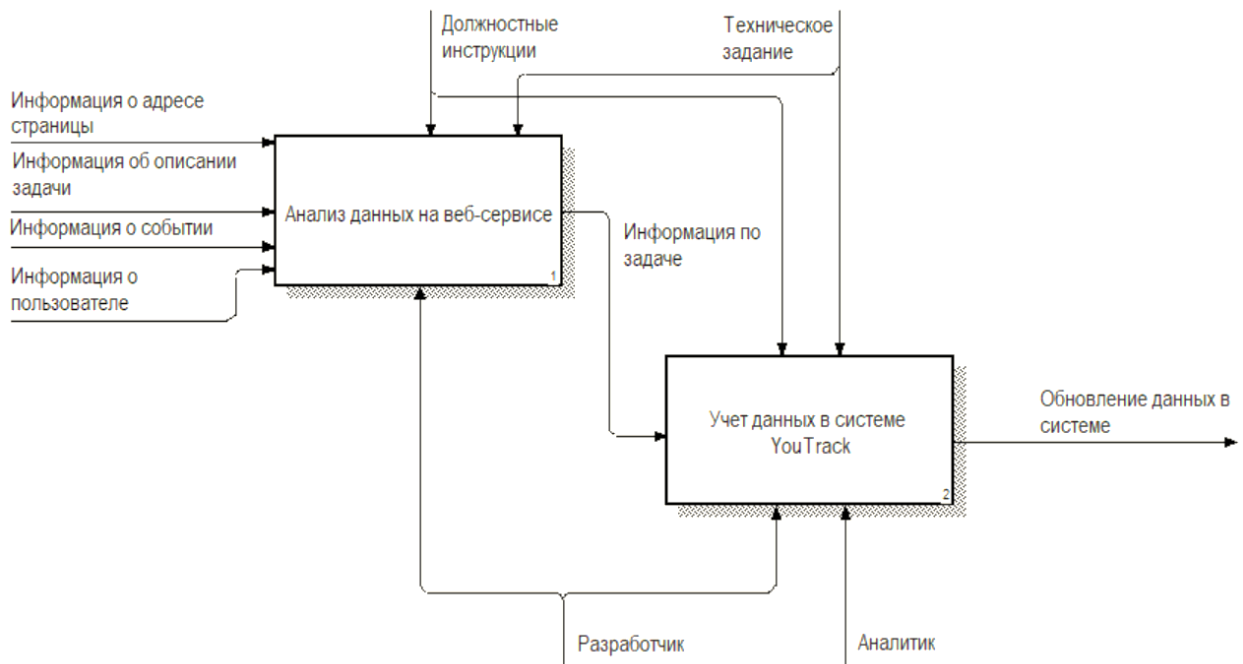


Figure 2.5 – Decomposition of the data integration process

The subsystem «Data Analysis on a Web Service» is the main subsystem of the integration module. Creating a functional model of the subsystem makes it possible to pin down exactly which information objects are used when executing business processes and individual operations performing.

The functional diagram is shown in Figure 2.6.

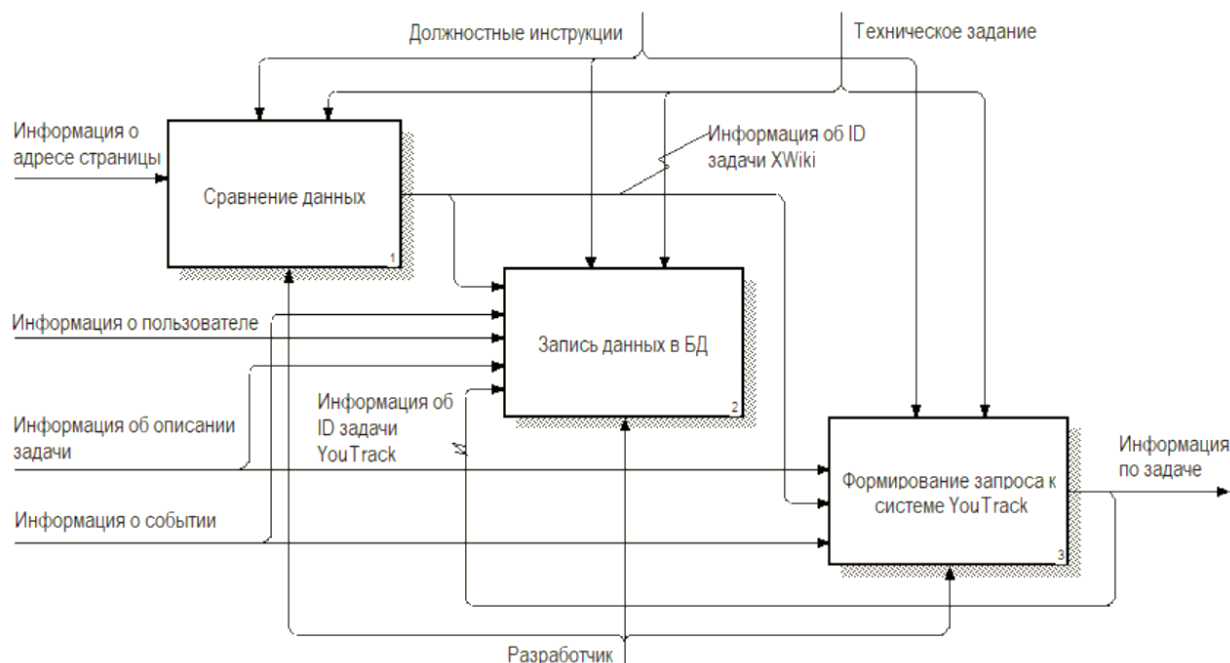


Figure 2.6 – Functional diagram

The Data Comparison feature allows you to correlate an XWiki task ID with a task ID in YouTrack. If such a pair is in the table, then the incoming data will update the task in the YouTrack system. A new record is added to the database in case if the ID could not be correlated. A task is created in YouTrack based on the XWiki task ID. The ID of the created task in YouTrack is recorded in the table of relations in the database of the integration module.

The "Write data to the database" function provides accounting and storage of information about users, the relationship between the task IDs of the two systems and the events that occurred in the XWiki system. Information about events is necessary for the functioning of the logging subsystem, which reflects all the information received.

The function «Creating a request to the YouTrack system» ensures the formation of a request to the YouTrack system based on all the received data from the

XWiki system. A request can be formed as for updating, adding and deleting data in the YouTrack system.

2.4 Designing a relational database model

The relational model is the simplest and most familiar form of presenting data in the form of tables.

The main advantage of the relational model is the simplicity of its support tools. The disadvantages include the stiffness of the data structure and the direct dependence of its speed on the size of the database. However, viewing the entire database for many operations represented in such a model may be necessary.

For the organization of the database will use a relational database. Therefore, it is necessary that the logical structure of the relational database be developed, because the solution of the problem will be carried out on its basis. We use the process approach to database development, determining the composition of only those data that are necessary to solve the problem.

The information in the database is represented by four relationships, which are described in table 2.3.

Table 2.3 – Accounting Information

relationship	Attribute	Description
Users	Code	User code
	Name	Username
Logs	Code	Journal code
	Correlation	Correlation code
	User	User code
	Date	Record date
	Event	Event name
Correlation	Code	Correlation code

	XWiki ID	Unique XWiki Requirement Key
	YouTrack ID	Unique YouTrack Requirement Key
Settings	Code	Setup Code
	YouTrack Address	YouTrack connection address
	E-mail	The email address to which you are authenticating when interacting with YouTrack
	YouTrack Password	Password used for authentication when interacting with YouTrack
	Login	Login to login to admin panel
	Password	Password to enter the administrative panel

The data model is a data schema of the subject area that is created to correctly display reality in the information system. Data is modeled to build a database.

The conceptual level of the database being developed is a summary presentation of the data. A conceptual domain model describes a logical data structure. It is the most complete presentation of data requirements by users of an information system. All entities, their attributes and domain connections are represented in the conceptual model. The entity-relationship diagram (ERD) is presented in Application A.

Primary, alternative and foreign keys of entities are shown, in addition to the names of entities and relationships, at the key level (KB - level). Specified properties of links (their cardinality and identifiability) are also indicated.

The HF-level diagram represents the logical structure of the relationships of the entities that make up the subject area. The conceptual model at the key level is presented in Application B for the described domain.

All entity attributes are shown at the attribute level (FA-level). This diagram includes the most complete definition of the structure of the developed system. The conceptual model at the attribute level for this subject area is presented in Application C.

2.5 Development of system requirements specification

2.5.1 System Requirements

Table 2.4 – groups of requirements

Symbol	Requirements group
F	General functional requirements
P	Requirements for integration tools
IS	Information Security Requirements
TS	Requirements for technical support
SR	Software Requirements
RG	Reliability requirements
D	Documentation Requirements

2.5.2 [F] General Functional Requirements

Table 2.5 – General Functional Requirements

Requirement Code	Requirements	Notes
F.01	Data processing requirements	
F.01.01	The system should add a link to the page with the XWiki requirement in the description of the YouTrack requirement.	

F.01.02	The system should compare the data from the XWiki system with the available data in YouTrack	
F.02	Data transfer requirements	
F.02.01	The system must provide the ability to send data to YouTrack	
F.02.02	The system must provide the ability to accept data from XWiki	
F.02.03	The system must provide the ability to accept data from YouTrack	
F.03	General requirements	
F.03.01	The system should create a task in YouTrack based on the data from XWiki	
F.03.02	The system should update the description of the YouTrack task based on the data received from XWiki	
F.03.03	The system should provide the provision of its functionality to other information systems through the API	
F.03.04	The system must fill in the rest of the default fields when creating a task in YouTrack	

2.5.3 [P] Requirements for integration tools

Table 2.6 – Requirements for integration tools

Requirement Code	Requirements	Notes
P.01	Requirements for integration with YouTrack	
P.01.01	The system must integrate the web service with YouTrack to enable data exchange.	
P.01.02	The system should interact with the web service and YouTrack via REST API YouTrack	
P.02	XWiki Integration Requirements	
P.02.01	The system must integrate the web service with XWiki to enable data exchange.	
P.02.02	The system should interact with the integration module and XWiki through the XWiki extension and web service API	

2.5.4 [IS] Information Security Requirements

Table 2.7 – Requirements for information security

Requirement Code	Requirements	Notes
IS.01	The system should support the implementation of a set of measures to ensure information security.	Protection against unauthorized access, violation of the integrity and reliability of stored and transmitted information
IS.02	Data confidentiality must be maintained in the system.	

2.5.5 [TS] Requirements for technical support

Table 2.8 – Requirements for technical support

Requirement Code	Requirements	Notes
TS.01	Web server requirements	
TS.01.01	16-core processor;	
TS.01.02	RAM: 8 GB	
TS.01.03	Disk space: 512 GB hard disk	
TS.01.04	Communication channel between servers: 1 Gbit/s	
TS.01.05	External communication channel: 100 Mbit/s	

2.5.6 [SR] Software Requirements

Table 2.9 – Software Requirements

Requirement Code	Requirements	Notes
SR.01	Operating System Requirements	
SR.01.01	The system must support operation in Microsoft Windows, Unix (Linux), Apple MacOS	
SR.02	Browser Requirements	
SR.02.01	Internet Explorer 8 and up (Windows only); Mozilla Firefox 13 and up; Safari 5 and up; Google Chrome 15 and up; Opera 11 and up.	

2.5.7 [RG] Reliability Requirements

Table 2.10 – Reliability Requirements

Requirement Code	Requirements	Notes
RG.01	System software should support restoring its operation when hardware is restarted.	
RG.02	The system should provide an increased level of information security in the event of accidents, technical failures (including loss of power)	
RG.03	The system must ensure the safety and integrity of previously entered data during updates	

2.5.8 [D] Documentation Requirements

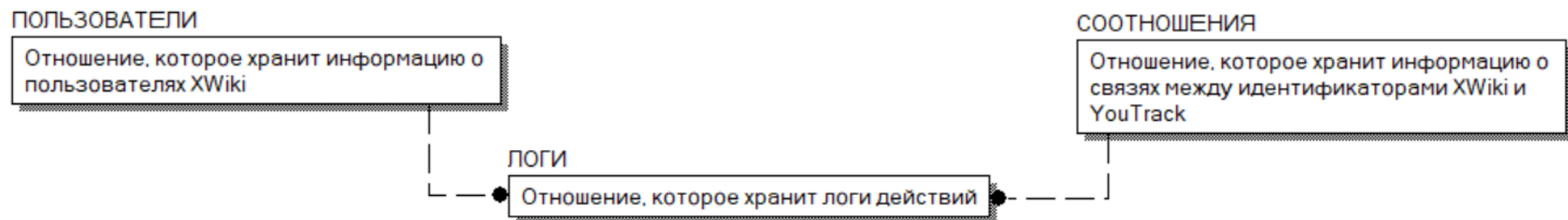
Table 2.11 – Documentation Requirements

Requirement Code	Requirements	Notes
D.01	Development of documentation on the API of the system should be carried out in accordance with this Technical Requirement.	The technical requirements include: System API documentation; Server distributions
D.02	Documentation must be in text format in Russian and submitted to the Customer in electronic form.	

ПРИЛОЖЕНИЕ Б

(Обязательное)

Инфологическая модель (Уровень определений)



ПРИЛОЖЕНИЕ В

(Обязательное)

Инфологическая модель (Уровень ключей)



ПРИЛОЖЕНИЕ Г

(Обязательное)

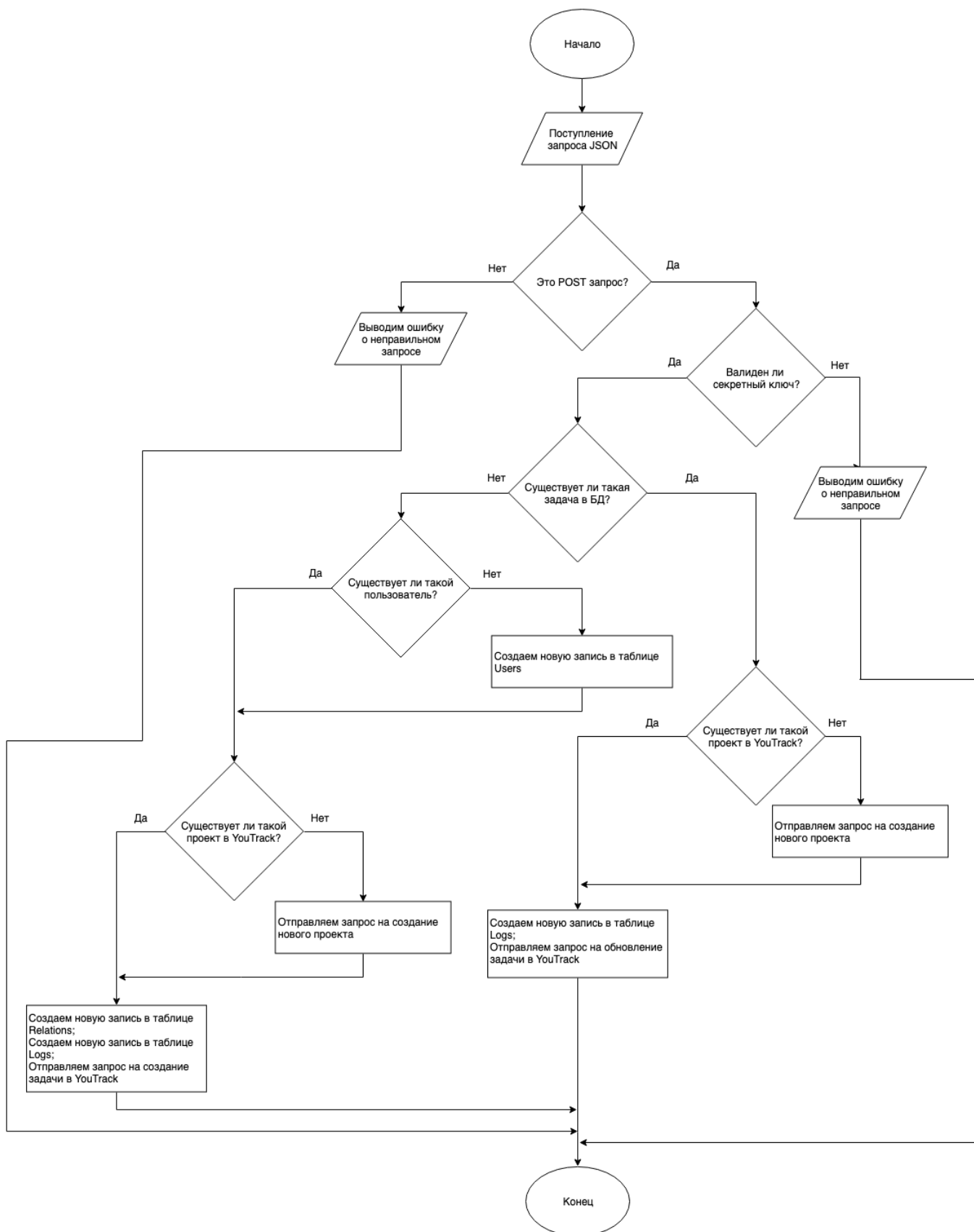
Инфологическая модель (Уровень атрибутов)



ПРИЛОЖЕНИЕ Д

(Обязательное)

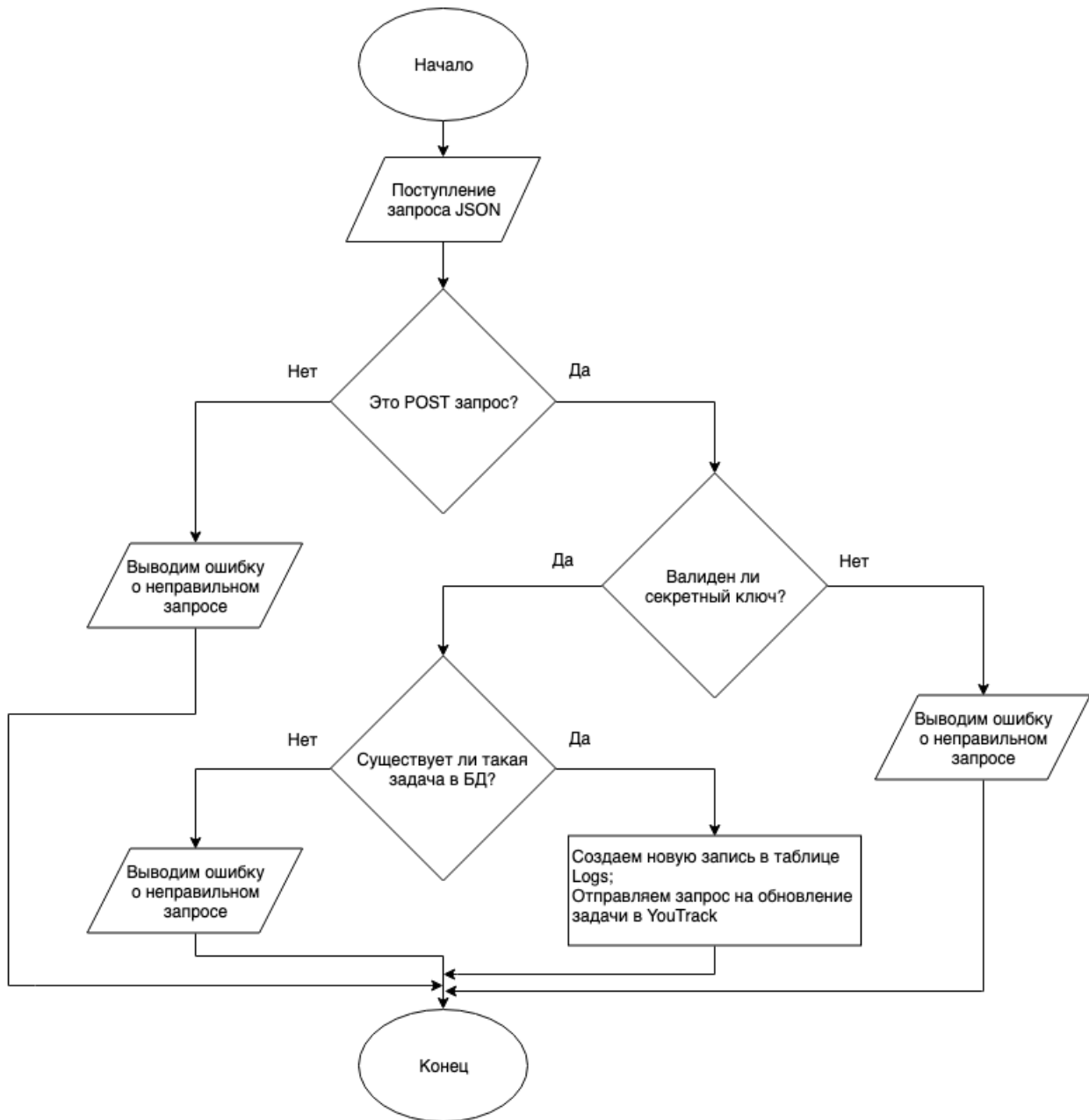
Алгоритм API метода «api/post/create»



ПРИЛОЖЕНИЕ Е

(обязательное)

Алгоритм API метода «api/post/update»



ПРИЛОЖЕНИЕ Ж

(обязательное)

Алгоритм API метода «api/post/delete»

