

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Научно-образовательный центр «Высшая ИТ школа»

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООИ
д-р. физ.-мат. наук, профессор

О.А.Змеев

подпись

« 11 » июня 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

СОЗДАНИЕ СИСТЕМЫ ПО ФОРМИРОВАНИЮ И СОГЛАСОВАНИЮ ПЕРВИЧНЫХ
УЧЕТНЫХ ДОКУМЕНТОВ ПО КОМАНДИРОВКАМ В ЭЛЕКТРОННОМ ВИДЕ НА
ПЛАТФОРМЕ «1С-Битрикс24»

по направлению подготовки 09.03.04 Программная инженерия
направленность (профиль) «Программная инженерия»

Трубников Владислав Игоревич

Руководитель ВКР
д-р. физ.-мат. наук, профессор

О.А. Змеев

подпись

« 10 » июня 2022 г.

Консультант ВКР
начальник управления цифровых
решений ТГУ

Д.А. Соколов

подпись

« 9 » июня 2022 г.

Автор работы
студент группы № 971810

В.И. Трубников

подпись

« 09 » июня 2022 г.

Томск – 2022

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
НОЦ «Высшая ИТ школа»

УТВЕРЖДАЮ
руководитель ООП
д-р. физ.-мат. наук, профессор
О.А.Змеев
« 21 » февраля 2022 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающегося
Трубникову Владиславу Игоревичу

(Ф.И.О. обучающегося)

по направлению подготовки Программная инженерия, направленность «Программная инженерия»

1. Тема выпускной квалификационной работы бакалавра

Создание комплексной системы по формированию и согласованию документов по командировкам в электронном виде на платформе «1С-Битрикс24»

2. Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис – « 09 » июня 2022 г.

б) в ГЭК – « 11 » июня 2022 г.

3. Исходные данные к работе:

Цель: разработать модуль для корпоративного портала «ТГУ.Битрикс24» по электронному документообороту командировок

цели и задачи ВКР, ожидаемые результаты

Задачи: анализ требований, проектирование архитектуры, разработка модуля

Ожидаемые результаты: система электронного документооборота командировок, соответствующая коллективным требованиям заказчика в лице ТГУ.

Организация, по тематике которой выполняется работа

НИИ ТГУ

Руководитель выпускной квалификационной работы

Д-р физ.-мат. наук, профессор
кафедры программной инженерии
ТГУ

(должность, место работы)

(подпись)

/ О.А.Змеев

(И.О. Фамилия)

Консультант выпускной квалификационной работы

Начальник управления цифровых
решений ТГУ

(должность, место работы)

(подпись)

/ Д.А. Соколов

(И.О. Фамилия)

Задание принял к исполнению

« 18 » 02 2022 г.
(дата)

(подпись)

/ В.И. Трубников

(И.О. Фамилия)

АННОТАЦИЯ

Выпускная квалификационная работа 59 стр., 22 рис., 5 табл., 16 листингов, 12 источников.

РНР, 1С-BITRIX24, КОРПОРАТИВНЫЙ ПОРТАЛ, ЭЛЕКТРОННЫЙ ДОКУМЕНТООБОРОТ

Цель работы – разработка модуля, предоставляющего функционал электронной регистрации командировок и авансовых отчётов, их согласования и генерации по ним основных и сопутствующих документов, для портала электронного документооборота ТГУ «ТГУ.Битрикс24»

Результат работы – проведен анализ задачи, полностью реализована и представлена для внедрения часть модуля для работы с командировками, формализованы первичные требования для части модуля по работе с авансовыми отчётами.

СОДЕРЖАНИЕ

Введение	4
1. Анализ и фиксация требований	6
1.1. Функциональные требования	6
1.2. Нефункциональные требования	8
1.3. Сценарии вариантов использования	10
2. Проектирование.....	16
2.1. Процесс согласования.....	16
2.2. Модель предметной области.....	18
3. Реализация	21
3.1. Bitrix Framework.....	21
3.2. Модули	22
3.3. Инфоблоки.....	26
3.4. Highload-блоки	29
3.5. Компоненты.....	33
3.6. Бизнес-процессы	45
3.7. Процессы, поставленные на стоп.....	52
3.8. Генерация документов.....	55
Заключение.....	57
Список литературы.....	58

ВВЕДЕНИЕ

«ТГУ.Битрикс24» — это портал для электронного документооборота в ТГУ, реализующий такие направления цифровизации, как электронная система управления проектами и сервис по рекрутингу иностранных абитуриентов. В рамках проекта, описываемого в данной выпускной квалификационной работе, передо мной стояла задача расширить его функциональность сервисом для электронной регистрации командировок и авансовых отчётов по ним, формирования по ним основных и сопроводительных документов и их согласования, позволяющим формировать имеющие юридическую силу решения.

Можно выделить несколько причин для перевода обработки командировок и авансовых отчётов в цифровой формат. Во-первых, автоматизация процесса позволяет существенно сократить затрачиваемое на него время сотрудников, так как минимизирует вероятность ошибок при первичном заполнении документа и при его согласовании, и избавляет командированное лицо или его представителя от необходимости переносить физический документ и вживую встречаться с согласующими. Последнее особо оптимально в связи с эпидемиологической обстановкой, которая на момент начала работы над проектом была значительно более серьёзной, чем сейчас. Кроме того, как и при любой цифровизации, обеспечивается более надёжное сохранение данных по завершённым документам и более удобный доступ к ним. Таким образом, приказом ректора ТГУ была установлена необходимость перехода на электронный документооборот по командировкам к началу 2023 года.

Перевод аналогового бизнес-процесса в электронный осуществлялся при взаимодействии с сотрудниками управления бухгалтерского учёта ТГУ, преимущественно в неформальном виде: после того, как мы получили первоначальную диаграмму деятельности для бизнес-процесса согласования, все последующие уточнения и дополнения добавлялись в устном или неформальном письменном виде. Я в ходе работы над проектом отвечал за все, кроме общения с обладающими компетенциями для определения электронного процесса согласования сотрудниками.

Задачи работы:

1. Создание структуры данных для хранения всей необходимой для формирования документов по командировкам и авансовым отчётам информации.

2. Создание форм, которые командированное лицо или его представитель будет использовать для ввода используемой для генерации документов информации.
3. Генерация первичных и дополнительных документов по командировкам и авансовым отчётам.
4. Реализация бизнес-процесса для согласования командировок и авансовых отчётов.

1. АНАЛИЗ И ФИКСАЦИЯ ТРЕБОВАНИЙ

1.1.Функциональные требования

Прежде всего, опишем основные существующие в системе роли и их варианты использования. Имеет смысл разбить все варианты использования на две группы: относящиеся к работе с командировками и относящиеся к работе с авансовыми отчётами. Несмотря на то, что полный электронный документооборот в реализованной системе предполагает работу и с командировкой, и с авансовым отчётом, в рамках одной командировки это два строго последовательных этапа.

В системе выделяются следующие роли для работы с командировками:

- **Инициатор командировки** – пользователь, заполняющий форму командировки и запускающий бизнес-процесс её согласования, а также ответственный за командировку после запуска бизнес-процесса согласования.
- **Командируемое лицо** – пользователь, по которому оформляется командировка. Может совпадать с инициатором командировки.
- **Согласующий** – руководитель подразделения/руководитель проекта/сотрудник отдела безопасности/сотрудник отдела кадров, подписывающий командировку простой электронной подписью.
- **Координатор** – пользователь, выбирающий руководителя организации для заверения командировки квалифицированной электронной подписью.
- **Руководитель организации** – владеющий квалифицированной электронной подписью проректор или ректор ТГУ, заверяющий командировку как руководитель организации.

Инициатор командировки должен иметь возможность:

- заполнить основную форму командировки;
- заполнить таблицу «Места и сроки командирования»;
- заполнить таблицу «Условия проезда»;
- заполнить таблицу «Условия проживания»;
- заполнить таблицу «Расчёт обязательств по командировке»;
- запустить процесс согласования;

- внести исправления в командировку;
- отменить командировку;
- скопировать существующую командировку;
- делегировать создание командировок другому пользователю;
- создать банковский реквизит;
- просмотреть командировку.

Командируемое лицо должно иметь возможность:

- делегировать создание командировок другому пользователю;
- создать банковский реквизит;
- просмотреть командировку.

Согласующий должен иметь возможность:

- просмотреть командировку;
- согласовать командировку;
- делегировать согласование командировки.

Координатор должен иметь возможность:

- выбрать проректора.

Руководитель организации должен иметь возможность:

- заверить командировку ЭЦП.

Из вышеперечисленных вариантов использования следующие являются архитектурно значимыми: «Заполнить основную форму командировки», «Заполнить таблицу» (любую из четырёх), «Внести исправления в командировку», «Согласовать командировку».

По работе с авансовыми отчётами выделяются следующие роли:

- **Инициатор** – пользователь, заполняющий формы авансового отчёта и запускающий бизнес-процесс согласования, а также ответственный за внесение в отчёт необходимых корректировок в процессе согласования.
- **Командируемое лицо** – пользователь, по которому заполняется авансовый отчёт.
- **Согласующий** – аналогично согласующему по командировкам.

- **Руководитель организации** – аналогично согласующему по командировкам.

Инициатор должен иметь возможность:

- создать черновик отчёта;
- заполнить основную форму отчёта;
- заполнить таблицу «Отчёт о расходах по командировке»;
- запустить процесс согласования;
- внести исправления в отчёт;
- отменить отчёт;
- делегировать создание отчётов другому пользователю.

Командируемое лицо должно иметь возможность:

- делегировать создание отчётов другому пользователю.

Согласующий должен иметь возможность:

- просмотреть отчёт;
- согласовать отчёт;
- делегировать согласование отчёта.

Руководитель организации должен иметь возможность:

- заверить отчёт ЭЦП.

Из вышеперечисленных вариантов использования архитектурно значимым является «Создать черновик отчёта». Остальное в целом покрывается архитектурой, необходимой для реализации вариантов использования по работе с командировками.

1.2. Нефункциональные требования

Можно выделить следующие нефункциональные требования, выявленные в ходе работы над проектом:

- Время открытия списка командировок, а также время запуска бизнес-процесса или его перезапуска после корректировки заявки должно составлять не более 2 секунд.

- Реализованные инструменты для редактирования бизнес-процесса согласования и сопутствующих подсистем (напр. подсистемы рассылки уведомлений о зависших командировках) должны предоставлять достаточные графические интерфейсы, чтобы использоваться администратором без навыков программирования.
- Система должна использовать минимальное количество внешних сервисов.
- Все используемые внешние сервисы должны быть доступны с компьютеров, расположенных во внутренних сетях ТГУ.
- Основные данные по командировке/авансовому отчёту и таблицы должны иметь возможность заполняться в произвольном порядке
- Просмотр таблицы по командировке/авансовому отчёту и добавление новых записей в неё/редактирование существующих записей должны происходить в едином слайдере без перезагрузок
- Для согласования командировка/авансовый отчёт должен быть представлен в специальной форме для просмотра, чтобы избежать необходимости прямой работы с документом
- Просмотр и согласование командировки/авансового отчёта должны происходить на одной странице
- Нанесение квалифицированной электронной цифровой подписи должно производиться в одно действие на стороне «ТГУ.Битрикс24»

1.3. Сценарии вариантов использования

Название ВИ	Заполнить основную форму командировки
Цель	Прохождение первого этапа в создании решения о командировании
Актер	Инициатор командировки
Предусловия	Пользователь авторизован, находится на главной странице сайта
Сценарий	<ol style="list-style-type: none">1. Пользователь нажимает на виджет «Командировки» в компоненте «ТГУ.Сервисы» и выбирает «Создать новую командировку»2. Система открывает страницу создания новой командировки3. Пользователь выбирает командируемое лицо4. Система устанавливает доступные для выбора должности и банковские реквизиты5. Пользователь выбирает должности, по которым командируемое лицо едет в командировку6. Система подставляет руководителей структурного подразделений7. Пользователь заполняет остальные поля и нажимает на кнопку «Сохранить»8. Система сохраняет командировку и присваивает ей статус «Черновик»9. Система обновляет страницу и открывает для заполнения 4 дополнительные таблицы

Таблица 1 (продолжение)

Альтернативы	<p>6.1 Пользователю нужно вручную выбрать других руководителей подразделения</p> <p>6.1.1 Пользователь нажимает иконку удаления на тегах с ненужными руководителями организации</p> <p>6.1.2 Система убирает из формы выбранных руководителей подразделения</p> <p>6.1.3 Пользователь использует нажимает на кнопку «Добавить»</p> <p>6.1.4 Система открывает компонент для выбора пользователей</p> <p>6.1.5 Пользователь выбирает нужных ему пользователей</p> <p>6.1.6 Система добавляет выбранных пользователей как руководителей подразделений</p> <p>7.1 Пользователь выбрал не все должности командидуемого лица</p> <p>7.1.1 Система отображает однократное сообщение с просьбой загрузить документы по отпуску</p> <p>7.2 Пользователь заполнил не все обязательные поля</p> <p>7.2.1 Система отображает сообщение об ошибке со списком незаполненных обязательных полей</p>
--------------	---

Таблица 2 – Сценарий варианта использования "Заполнить таблицу «Места и сроки командирования»"

Название ВИ	Заполнить таблицу «Места и сроки командирования»
Цель	Заполнение одной из обязательных таблиц
Актер	Инициатор командировки
Предусловия	Пользователь находится в форме командировки, открытой для редактирования

Таблица 2 – (продолжение)

Сценарий	<ol style="list-style-type: none"> 1. Пользователь нажимает на кнопку «Добавить» в соответствующей грид-таблице 2. Система открывает слайдер с формой создания элемента таблицы 3. Пользователь выбирает страну командирования 4. Система определяет тип поля «Населенный пункт» 5. Пользователь указывает населенный пункт 6. Система определяет формат и автозаполнение нормативов 7. Пользователь указывает норматив и заполняет даты командировки 8. Система рассчитывает продолжительность и подставляет значения в поля «Продолжительность» 9. Пользователь вводит остальные поля и нажимает на кнопку «Сохранить» 10. Система сохраняет элемент таблицы 11. Система обновляет отображение таблицы на странице
Альтернативы	<ol style="list-style-type: none"> 7.1 Дата начала/выезда позже даты окончания/приезда <ol style="list-style-type: none"> 7.1.1 Система отображает alert с ошибкой 7.1.2 Система очищает даты 8.1. Пользователь пытается заполнить поле «Продолжительность» вручную <ol style="list-style-type: none"> 8.1.1 Система отображает alert с ошибкой 9.1 Пользователь заполнил не все обязательные поля <ol style="list-style-type: none"> 9.1.1 Система выводит список незаполненных обязательных полей 11.1 Пользователь хочет добавить дополнительные элементы в таблицу <ol style="list-style-type: none"> 11.1.1 Шаг 1. 11.2 Пользователь хочет удалить созданный элемент

Таблица 2 (продолжение)

Альтернативы	<p>11.2.1 Пользователь нажимает на нужный элемент и выбирает «Удалить»</p> <p>11.2.2 Система удаляет выбранный элемент</p> <p>11.2.3 Шаг 10.</p> <p>11.3 Пользователь хочет отредактировать созданный элемент</p> <p>11.3.1 Пользователь нажимает на нужный элемент и выбирает «Редактировать»</p> <p>11.3.2 Система открывает слайдер, где в поля подставлены текущие значения</p> <p>11.3.3 Шаг 3.1 / 4</p>
--------------	---

Таблица 3 – Сценарий варианта использования «Внести исправления в командировку»

Название ВИ	Внести исправления в командировку
Цель	Перезапуск согласования после отклонения заявки
Актер	Инициатор командировки
Предусловия	Пользователь находится в форме командировки со статусом «На исправлении»
Сценарий	<ol style="list-style-type: none"> 1. Система отображает пользователю сообщение от отклонившего согласующего. 2. Пользователь вносит изменения в основную форму и нажимает кнопку “Сохранить”. 3. Система обновляет элемент командировки. 4. Пользователь исправляет содержимое 4 таблиц (аналогично шагам 10.1.1, 10.2.1, 10.3.1 предыдущего сценария) и нажимает на кнопку «Запустить» 5. Система перезапускает бизнес-процесс согласования

Таблица 4 – сценарий варианта использования «Согласовать командировку»

Название ВИ	Согласовать командировку
Цель	Проставление простой электронной подписи согласующего в документе командировки
Актер	Согласующий
Предусловия	Пользователь авторизован
Сценарий	<ol style="list-style-type: none"> 1. Система отображает уведомление с просьбой согласовать командировку и сохраняет его в списке уведомлений 2. Пользователь переходит по ссылке из уведомления 3. Система отображает страницу задания на согласование, где представлены нужные данные по командировке, список файлов по ней и список причин предыдущих отклонений, при наличии таковых, а также форма согласования 4. Пользователь выбирает должность, по которой согласует, из списка и нажимает на кнопку «Согласовать» 5. Система выводит страницу выполненного задания БП
Альтернативы	<ol style="list-style-type: none"> 3.1 У пользователя только одна активная должность <ol style="list-style-type: none"> 3.1.1 В форме согласования не отображается селектор должности. 3.1.2 Пропуск шага 4. 4.1 В данных по командировке содержатся ошибки <ol style="list-style-type: none"> 4.1.1 Пользователь вводит комментарий с перечислением ошибок и нажимает кнопку «Отклонить» 4.1.2 Система выводит страницу выполненного задания БП

Таблица 5 – сценарий варианта использования «Создать авансовый отчёт»

Название ВИ	Создать авансовый отчёт
Цель	Создание черновика авансового отчёта
Актер	Инициатор
Предусловия	Пользователь находится на странице списка командировок
Сценарий	<ol style="list-style-type: none"> 1. Пользователь выбирает «Авансовый отчёт» в выпадающем списке действий по командировке 2. Система создаёт авансовый отчёт по командировке в статусе «Черновик» 3. Система автоматически заполняет таблицу «Отчёт о выполненной работе в командировке» на основании данных из таблицы «Места и сроки командирования» в командировке 4. Система переводит пользователя на страницу созданного авансового отчёта
Альтернативы	<ol style="list-style-type: none"> 2.1 По выбранной командировке уже существует авансовый отчёт <ol style="list-style-type: none"> 2.1.1 Система переводит пользователя на страницу существующего авансового отчёта

2. ПРОЕКТИРОВАНИЕ

2.1. Процесс согласования

После создания заявки на командировку по ней должен запускаться бизнес-процесс согласования (подробнее о деталях реализации бизнес-процессов в Bitrix Framework рассказано в части 3.6). Бизнес-процесс должен выполнить те же функции, что и аналоговый процесс согласования. Во-первых, он используется для сбора подписей: простых электронных подписей (то есть подписей, надежность которых обеспечивается просто авторизацией через логин и пароль) и квалифицированной электронной подписи руководителя организации – подписи с использованием методов криптографической защиты, предоставляемой вне портала «ТГУ.Битрикс24». Во-вторых, он используется для сбора дополнительных данных по командировке: в ходе согласования бухгалтер и сотрудник планового управления должны заполнить недоступные для пользовательского редактирования таблицы «Справочная информация» и «Финансовое обеспечение».

Бизнес-процесс состоит из следующих этапов:

- **Создание заявки** – здесь происходит сбор впоследствии используемой информации о согласующих, выбор научного или общего пути согласования и генерация первичного документа для показа руководителю структурного подразделения.
- **Руководитель подразделения, Управление безопасности, Руководитель проекта, Отдел кадров** – содержимое статусов практически идентично: определяется, проходила ли раньше попытка согласования в этом статусе, и согласующему (или согласующим) отправляется задание на согласование с информацией о командировке и набором сгенерированных и загруженных пользователем документов; в случае, если попытка согласования в статусе уже была, задание также содержит список комментариев к предыдущим отклонениям документа и измененное описание.
- **В бухгалтерии** – в этом статусе документ проходит последовательное согласование с бухгалтером и сотрудником ПФУ на внешнем сервисе «ТГУ.Контрагенты». В результате согласования здесь проставляются две подписи, а также заполняются таблицы «Справочная информация» и «Финансовое обеспечение».

- **Координатор** – зависит от выбираемого при изначальном создании черновика заявки направления командировки. Если направление общее, координатор выбирает проректора, который должен будет заверить набор файлов по командировке квалифицированной ЭЦП, и статус меняется на «Руководитель организации». Если направление научное, координатор сам заверяет набор файлов квалифицированной ЭЦП.
- **Руководитель организации** – руководитель организации заверяет набор файлов квалифицированной ЭЦП.
- **Завершено** – командируемому лицу отправляется уведомление об успешном согласовании.
- Если на каком-то из этапов согласования заявка будет отклонена, бизнес-процесс переходит в статус «**Корректировка заявки**». Командируемому лицу отправляется уведомление о необходимости откорректировать командировку и запустить повторное согласование с выводом отклонившего и его комментария. После исправления заявки процесс согласования откатывается к более раннему статусу.

представленная ниже модель предметной области для работы с командировками.

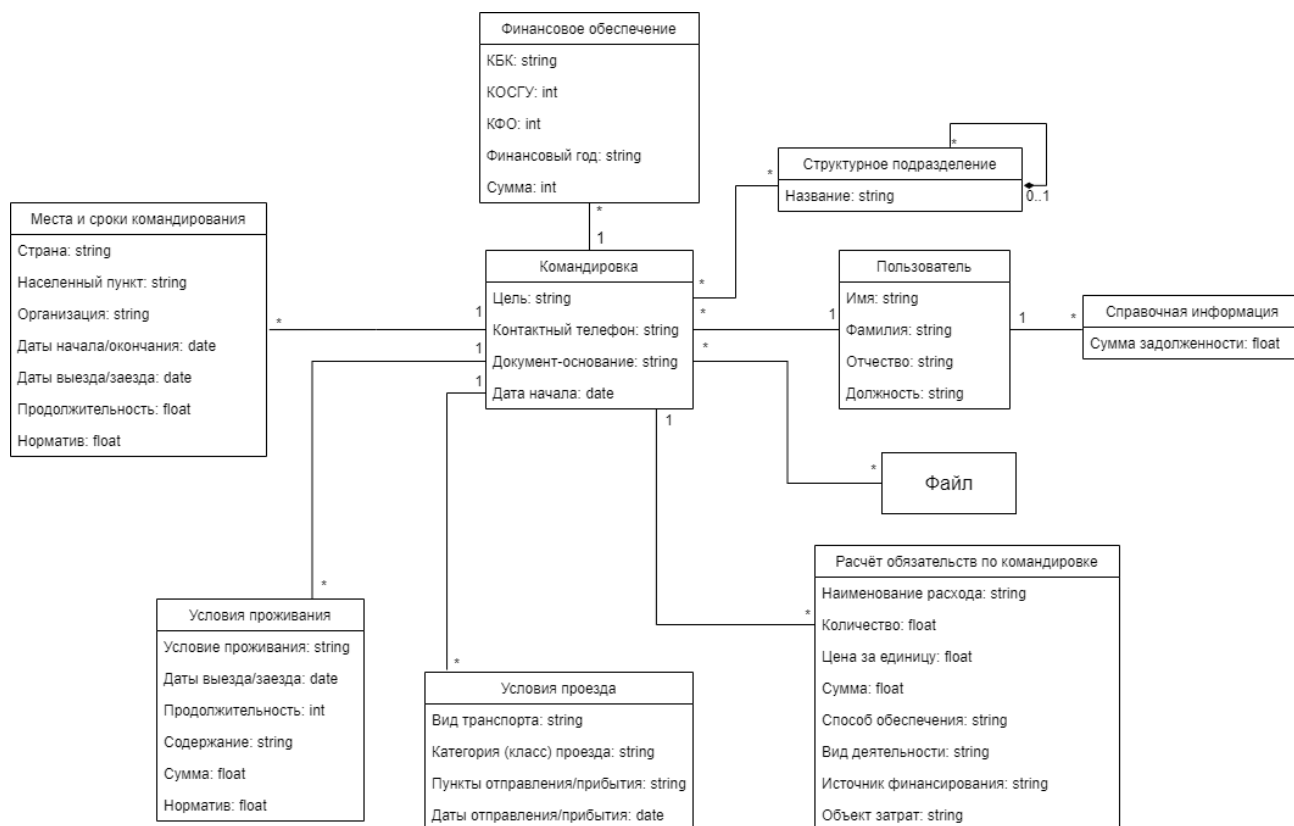


Рисунок 2 – модель предметной области по командировке

Первоначально пользователем заполняются «Командировка» - основная таблица, содержащая такую информацию, как командированное лицо, цель командировки, должности, по которым лицо едет в командировку, и т.д. - и 4 дополнительные таблицы, также участвующие в генерации документа:

1. «Места и сроки командирования»
2. «Условия проезда»
3. «Условия проживания»
4. «Расчёт обязательств по командировке»

После первого согласования в бухгалтерии на стороне «ТГУ.Контрагенты» заполняются таблицы «Финансовое обеспечение» и «Справочная информация» (задолженности пользователя по предыдущим командировкам).

4 заполняемые пользователем таблицы и таблица «Финансовое обеспечение» создаются исключительно в процессе согласования конкретной заявки и не выполняют никаких функций

вне него. Сущности «Структурное подразделение», «Пользователь» и «Файл» существуют независимо от командировки и реализованы в Bitrix Framework по умолчанию.

По результатам рассмотрения шаблона документа авансового отчёта и согласования процесса его согласования с сотрудниками из «ТГУ.Контрагенты» для работы с авансовыми отчётами была принята следующая модель предметной области.

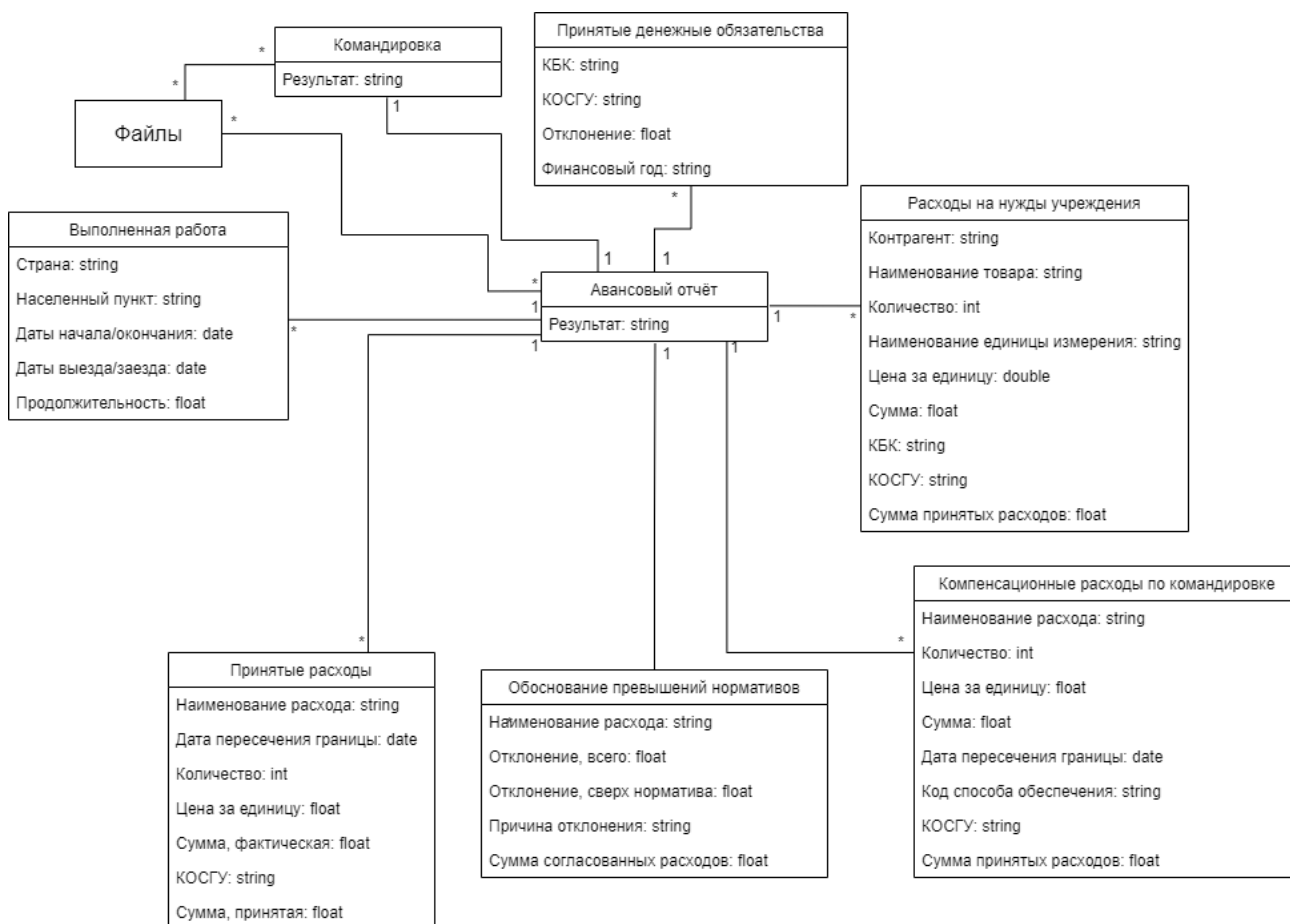


Рисунок 3 – модель предметной области авансового отчёта

При работе с авансовым отчётом пользователем заполняются таблицы «Расходы на нужды учреждения», «Компенсационные расходы по командировке», «Выполненная работа» и «Обоснование превышений нормативов».

Важным моментом является то обстоятельство, что авансовый отчёт, в отличие от командировки, практически не имеет собственных полей, а также не связывается напрямую с такими сущностями, как «Пользователь» и «Структурное подразделение». Вместо этого каждый авансовый отчёт содержит ссылку на командировку, по которой он заполняется, и существенная часть информации подставляется оттуда.

3. РЕАЛИЗАЦИЯ

3.1. Bitrix Framework

Bitrix Framework – это созданная на основе PHP платформа для разработки веб-приложений. На этой платформе компанией 1С-Битрикс создан используемый в решении продукт 1С-Битрикс24: Корпоративный портал». Bitrix Framework обеспечивает маршрутизацию; по умолчанию реализует ряд таблиц БД, такие, например, как пользователи и сообщения, и предоставляет графический интерфейс для управления данными; предоставляет ORM; реализует дополнительный уровень абстракции над таблицами СУБД в виде инфоблоков и highload-блоков; предоставляет пользователю большое количество готовых компонентов и модулей (см. ниже) для быстрой реализации типовых решений, а также набор классов и функций для облегчения работы программирования на PHP в целом (например, набор высокоуровневых функций по обработке даты и времени) и JS-библиотеку.

В отличие от других популярных PHP-фреймворков, таких как Symfony и Laravel, Bitrix Framework не предлагает использовать какую-то определенную архитектуру. Согласно официальному курсу по Bitrix Framework, фреймворк использует архитектуру MVC в своих компонентах: в роли модели выступает API для работы с данными, в роли контроллера выступает сам компонент, а в роли представления – его шаблоны.

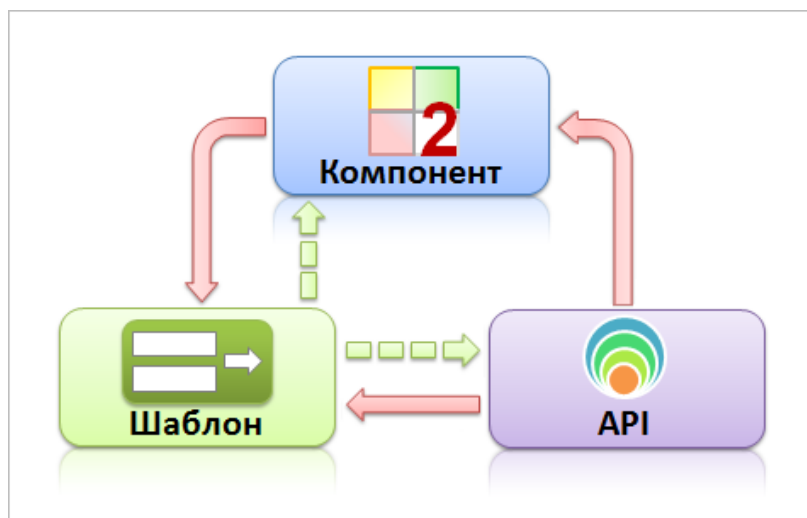


Рисунок 4 — Архитектура MVC в компонентах Bitrix Framework [1]

Такая трактовка архитектуры несколько отличается от классической – например, API по взаимодействию с данными расположено в модулях Битрикс, а не в компонентах, соответственно модель располагается в отдельной от контроллера и представления части системы; в любом случае, Bitrix Framework не требует организации кода в компоненты.

Различные решения на базе Bitrix Framework различаются количеством предоставляемых и поддерживаемых модулей, а также ограничениями на количество пользователей.

3.2. Модули

«Bitrix Framework имеет модульную структуру. Каждый модуль отвечает за управление определенными элементами и параметрами сайта: информационным наполнением и структурой сайта, форумами, рекламой, рассылкой, распределением прав между группами пользователей, сбором статистики посещений, оценкой эффективности рекламных кампаний и т.д.

Модуль – это блок, отвечающий за определенную функциональность продукта. Модуль может содержать API всех уровней (включая API доступа к данным, бизнес-логику и API пользовательского интерфейса), HTML верстку, пользовательские интерфейсы, компоненты, роботы, административные страницы и тому подобное.» [2]

В целом модули работают независимо друг от друга, однако в некоторых случаях одни используют для своей работы функционал других: например, модуль Торговый каталог предоставляет дополнительный функционал по работе с элементами инфоблока «Товарный каталог CRM» и в процессе, естественно, использует функционал модуля «Информационные блоки».

Хотя «1С-Битрикс: Корпоративный портал» предоставляет, в зависимости от лицензии, определенный набор стандартных модулей, возможны создание и установка собственных модулей. Гайдлайны Bitrix рекомендует организовывать выполняющий единые группы задач код в модули и регистрировать их, так как их установка позволяет использовать ряд возможностей:

- После установки модуля возможно использование класса `Bitrix\Main\Loader` для автоматического подключения всех его файлов, для которых обозначено указанное при установке модуля пространство имён. Это позволяет улучшить

читаемость кода и сокращает работу по подключению нужных файлов во внешних скриптах.

Листинг 1 – подключение модуля

```
\Bitrix\Main\Loader::includeModule('tsu.bustrip');
```

- Для модулей есть возможность использовать класс \Bitrix\Main\Config\Option, чтобы записывать для них в специальную таблицу произвольные настройки со строковым типом данных, что позволяет в одном месте собрать все меняющиеся используемые в модуле параметры, например URL. Настройки модулей также сокращают время, требуемое на перенос файлов между сайтами, при котором требуется заменять такие параметры как ID используемых в модуле инфоблоков и бизнес-процессов.

Листинг 2 – использование настроек модуля

```
$data = $this->getHLData(Option::get("tsu.bustrip",  
"IDIBlockPlaceTime"));
```

- В модулях возможно регистрировать события. Зарегистрированные в модуле события будут вызываться в скрипте тогда и только тогда, если в нём предварительно был подключён рассматриваемый модуль.

Листинг 3 – регистрация события в модуле

```
$eventManager = \Bitrix\Main\EventManager::getInstance();  
$eventHandler->registerEventHandler("iblock", "OnAfterIBlockElementAdd",  
self::MODULE_ID, "Handler", "OnAfterIBlockElementAddHandler");
```

В рамках разработки был создан один пользовательский модуль – «tsu.bustrip». В нём заключена основная часть бизнес-логики, используемой при создании заявок на командирование, их просмотре и в процессе их согласования. Ниже подробнее рассмотрены его файлы и настройки.

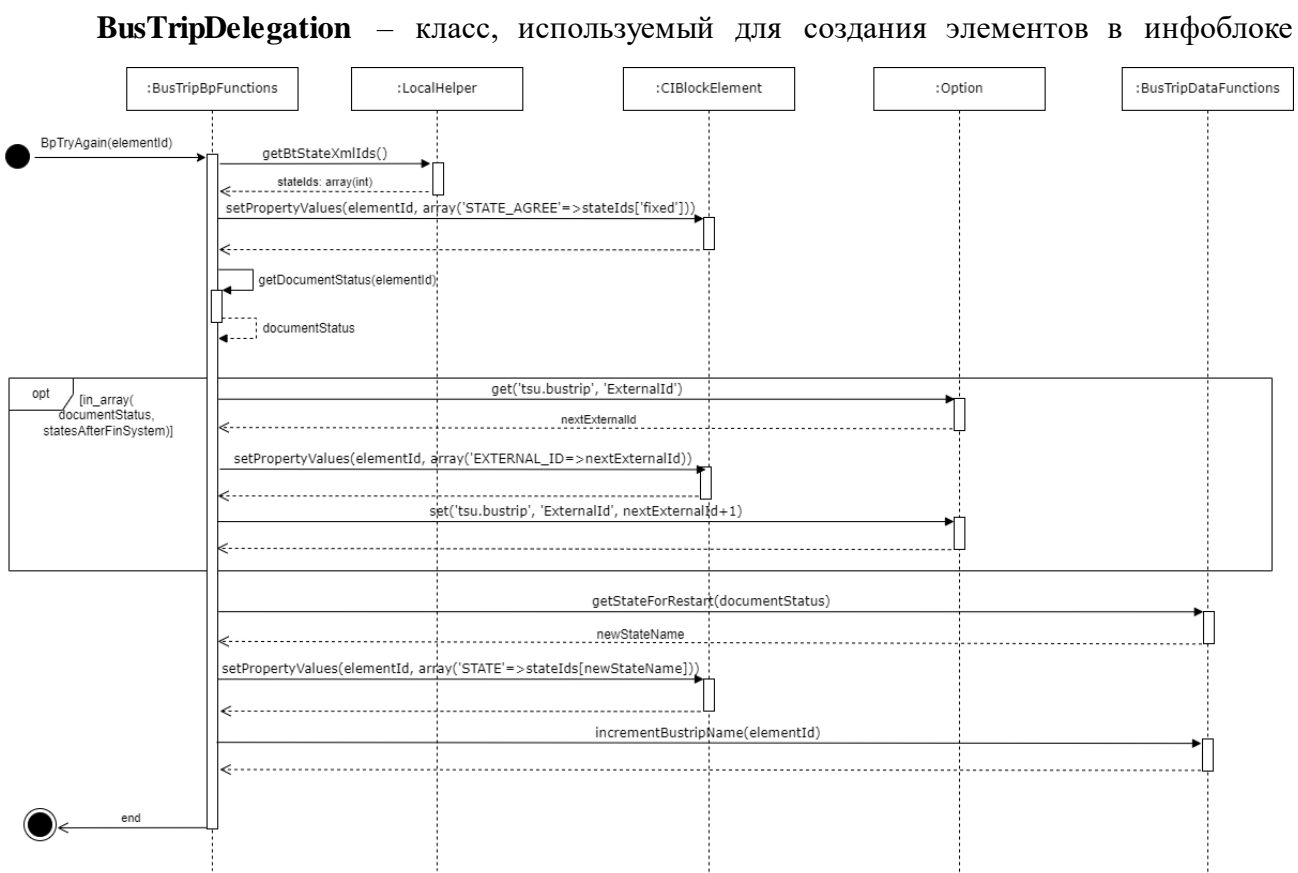
BusTripDataFunctions – статический класс, используемый для работы с элементами инфоблока «Командировка на территории РФ» и связанными highload-блоками. Содержит функции для копирования заявок, добавления подписей в элемент инфоблока при согласовании, и создания архивов документов для определенных этапов согласования.

BusTripBpFunctions – класс, используемый для работы с бизнес-процессом согласования командировки. Содержит функции для запуска, отмены и копирования бизнес-

процессов, работу с их логами (загрузка лога и комментариев согласующих для отображения в форме командировки, копирование лога при перезапуске бизнес-процесса, установление последнего достигнутого статуса согласования), и создания архивов документов для определенных этапов согласования.

На рисунке представлена диаграмма последовательности для функции, обрабатывающей перезапуск бизнес-процесса согласования, которую согласующий вызывает после внесения требуемых изменений в отклоненную заявку

Рисунок 5 – обработка перезапуска бизнес-процесса согласования



делегирований и получения информации о текущих активных делегированиях.

BusTripManager – класс, используемый для обмена данных с финансовой системой ТГУ. Содержит функции для авторизации на портал «ТГУ.Контрагенты», отправки туда данных о командировках для прохождения согласования с бухгалтером и сотрудником планово-финансового управления, отправки данных об отменённых и аннулированных командировках, а также получения и обработки ответных данных. Использует два разных HTTP-клиента для GET и POST запросов; чтобы гарантировать возможность доступа к ним и избежать создания лишних клиентов, используются классы-синглтоны, содержащие соответствующим образом настроенный HTTP-клиент и реализующие нужные запросы.

BusTripFinalManager – класс, наследующийся от **BusTripManager** и используемый для отправки командировок на второе, финальное согласование на «ТГУ.Контрагенты» и обработки ответов.

ProjectManager – класс, используемый для загрузки данных об актуальных проектах и обновления данных в соответствующем highload-блоке.

GeneratorReport – класс, используемый для генерации Word- и PDF-документов по командировке из представленных шаблонов.

FileLinkCreation – статический класс, используемый для получения абсолютных ссылок на файлы по указанному полю командировки и создания из них гиперссылок для вывода в задании БП.

Для большинства классов «BusTrip» созданы аналогичные «Report»-классы для работы с авансовыми отчётами.

В настройках модуля указаны:

- ID инфоблока «Командировка на территории РФ»
- ID бизнес-процесса согласования командировки
- ID всех связанных с инфоблоком highload-блоков, таких как highload-блоки для заполнения таблиц в документе, highload-блок с данными о делегированиях, highload-блок с информацией банковских счетов и др.
- ID соответствующих таблицам в документе highload-блоков, заполнение которых обязательно перед отправкой заявки на согласование.
- Текущие значения присваиваемых в рамках автоинкремента номера командировки и ID для отправки в фин. систему ТГУ (только для чтения).
- Токен для авторизации запросов в фин. систему и время его истечения.
- Отображаемые в форме в полоске статусов согласования статусы и их цвета

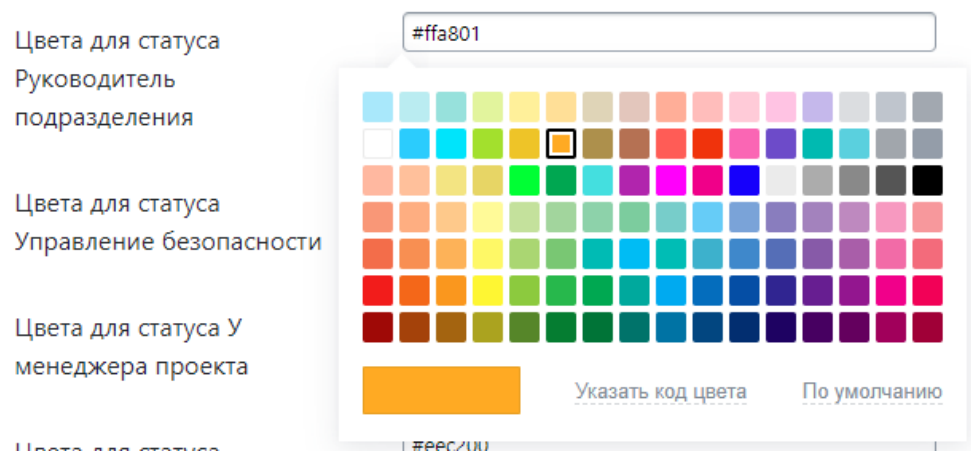


Рисунок 6 – выбор цвета для статуса «Руководитель подразделения» в настройках модуля

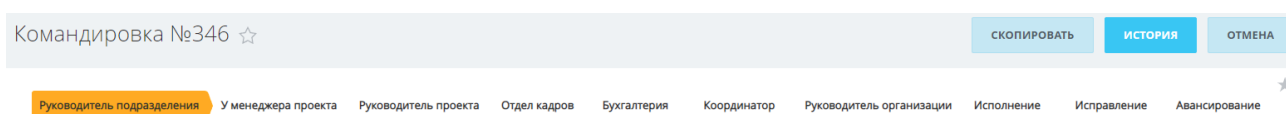


Рисунок 7 – полоска статусов в форме командировки

3.3.Инфоблоки

Информационные блоки (инфоблоки) – это один из основных модулей в Битрикс24, на основе которого работает ряд стандартных компонентов и других модулей. Модуль «Информационные блоки» является высочайшим реализованным в Bitrix Framework слоем абстракции баз данных. Вместо традиционного подхода к использованию реляционной базы данных для хранения данных инфоблоки реализуют следующую базовую структуру:

- в отдельной таблице (b_iblock_type) хранятся типы информационных блоков – категоризация инфоблоков по месту и способу использования;
- в отдельной таблице (b_iblock) хранятся основные данные о самих инфоблоках («таблицах»), такие как их название, их коды, на каких сайтах они расположены;
- в отдельной таблице (b_iblock_property) хранятся дополнительные заданные пользователем свойства конкретных инфоблоков – их название, тип данных, множественность, обязательность;
- в отдельной таблице (b_iblock_element) хранятся основные данные об элементах инфоблоков, такие как название, дата изменения, создатель, какому инфоблоку принадлежит элемент;

- в отдельной таблице (b_iblock_element_property) хранятся значения дополнительных свойств в элементах инфоблоков.

И так далее – иными словами, инфоблоки реализуют сильно модифицированный паттерн Entity-Attribute-Value для работы с данными. Если верить официальному курсу по Bitrix Framework [3], информационные блоки для своей реализации используют 4 таблицы базы данных, но в действительности для реализации даже самых основных их возможностей уже используется 5, и описания ряда участвующих в реализации системы инфоблоков таблиц были выше опущены – полная функциональность инфоблоков обеспечивается 27 таблицами, не считая дополнительных, которые другие модули создают для расширения функционала инфоблоков.

Естественно, у такого подхода есть недостатки, основным из которых является более низкая по сравнению с работой с классическими таблицами производительность, - основной причиной является сама необходимость обращения к большому количеству таблиц для составления SQL-запросов, и она усугубляется относительно плохой оптимизацией построителя запросов.[4] Среди достоинств модуля – децентрализованность структуры данных позволяет сделать пользовательский интерфейс максимально удобным: инфоблоки можно организовывать в группы, внутри них можно создавать разделы, можно менять типы данных без их потери. Классы модуля предоставляют возможности для экспорта и импорта данных и управления пользовательскими правами доступа. Факт разбиения данных на множество таблиц позволяет выделить в ORM ряд узкоспециализированных классов для интуитивно понятной работы с данными – специальные классы для управления инфоблоками, их разделами, элементами, пользовательскими полями, списочными полями и так далее. Наконец, главное достоинство (или по крайней мере, главная причина использовать инфоблоки) – уровень интеграции модуля с другими: например, модуль бизнес-процессов (см. ниже) позволяет в своих заданиях на с помощью пользовательского интерфейса менять элементы инфоблоков и получать доступ к значениям их свойств.

Ввиду низкой производительности инфоблоков по сравнению с обычными таблицами в БД или highload-блоками, более низкоуровневым слоем абстракции над БД в Bitrix, только основные таблицы проекта – «Командировки» и «Авансовый отчёт» – были созданы с их использованием. Главной причиной для реализации этой таблицы именно на инфоблоках являлась поддержка инфоблоков модулем «бизнес-процессы»: в проекте требуется возможность поддержки и модификации БП администратором без навыков программирования, соответственно, обязательным требованием является возможность

создания и изменения действий БП, работающих со свойствами элементов, через графический интерфейс.

Рисунок 8 – Действие бизнес-процесса «изменение полей документа» позволяет через графический интерфейс изменить свойство элемента инфоблока.

Ниже описаны отсутствующие в MySQL (база данных, используемая на сайте) типы данных, использованные в инфоблоке:

- Поле типа «Привязка к сотруднику» (напр. «Командируемое лицо») - хранит строку с ID сотрудника, в пользовательском интерфейсе для элемента инфоблока позволяет выбрать пользователя в селекторе с поиском и отображает его ФИО.
- Поле типа «Список» (напр. «Статус исправления») - хранит строку с ID выбранного значения в отдельной таблице, простой селектор в пользовательском интерфейсе. Значения списка, обозначенные для конкретного свойства конкретного инфоблока, могут быть выбраны только для этого поля в этом инфоблоке.
- Поле типа файл (напр. «Решение о командировании (PDF)») – хранит строку с ID зарегистрированного в отдельной таблице файла, в пользовательском интерфейсе использует надстройку над файловым input.

Листинг 4 – получение ID раздела инфоблоков

```
$department = \CIBlockSection::GetList(Array(), ['IBLOCK_ID'=>5,
'UF_PERSONA_CODE'=>$dept], false, ['ID'])->GetNext()['ID'];
```

Листинг 5 – изменение значения списочного свойства FINAL_FIN_ID без сброса остальных свойств

```
CIBlockElement::SetPropertyValuesEx($id, Option::get("tsu.bustrip",
"IDiblock"), $finId, "FINAL_FIN_ID");
```

Листинг 6 – установка названия для элемента \$newDocumentID без изменения пользовательских свойств

```
$el = new \CIBlockElement;  
$el->Update($newDocumentID, array('NAME'=>"Командировка №".$newFakeNumber));
```

3.4.Highload-блоки

Highload-блоки – инструмент абстракции над базой данных, использующий для своей реализации нормальные таблицы реляционной БД и одну дополнительную таблицу, соотносящую название таблицы БД с названием ORM-класса. Все ORM-классы, создаваемые в рамках модуля, наследуются от абстрактного класса DataManager и по умолчанию реализуют одинаковый набор функций, различаясь только в используемой таблице БД и в описании возвращаемой сущности (или наборе полей возвращаемого массива, если при работе не используются функции работы с автогенерируемыми моделями). Возможно доопределение сущности для конкретного инстанса класса-наследника DataManager.

Highload-блоки обеспечивают достаточно высокий уровень производительности, с ними просто работать через ORM, и при этом для них доступна возможность просмотра/редактирования в графическом интерфейсе через административную панель. Соответственно, через highload-блоки были реализованы все сущности в проекте, для которых не требовалась функциональность запуска над их элементами бизнес-процессов Bitrix.

Листинг 7 – получение класса для работы с конкретной таблицей

```
$hlblock = \Bitrix\Highloadblock\HighloadBlockTable::getById($hlBlockId) -  
>fetch();  
$entity = \Bitrix\Highloadblock\HighloadBlockTable::compileEntity($hlblock);  
$entity_data_class = $entity->getDataClass();
```

Листинг 8 – получение элементов делегирования, где делегат равен \$user, и крайний срок делегирования либо не наступил, либо не назначен

```
$rsData = $entity_data_class::getList(array(  
    'select' => array('*'),  
    'filter' => array(  
        'UF_DELEGATEE'=>$user,  
        array(  
            'LOGIC' => "OR",  
            array('>=UF_END' => ConvertTimeStamp(false, "SHORT")),  
            array('UF_END' => false)  
        )  
    ));
```

Ниже представлена схема реализованных таблиц БД.

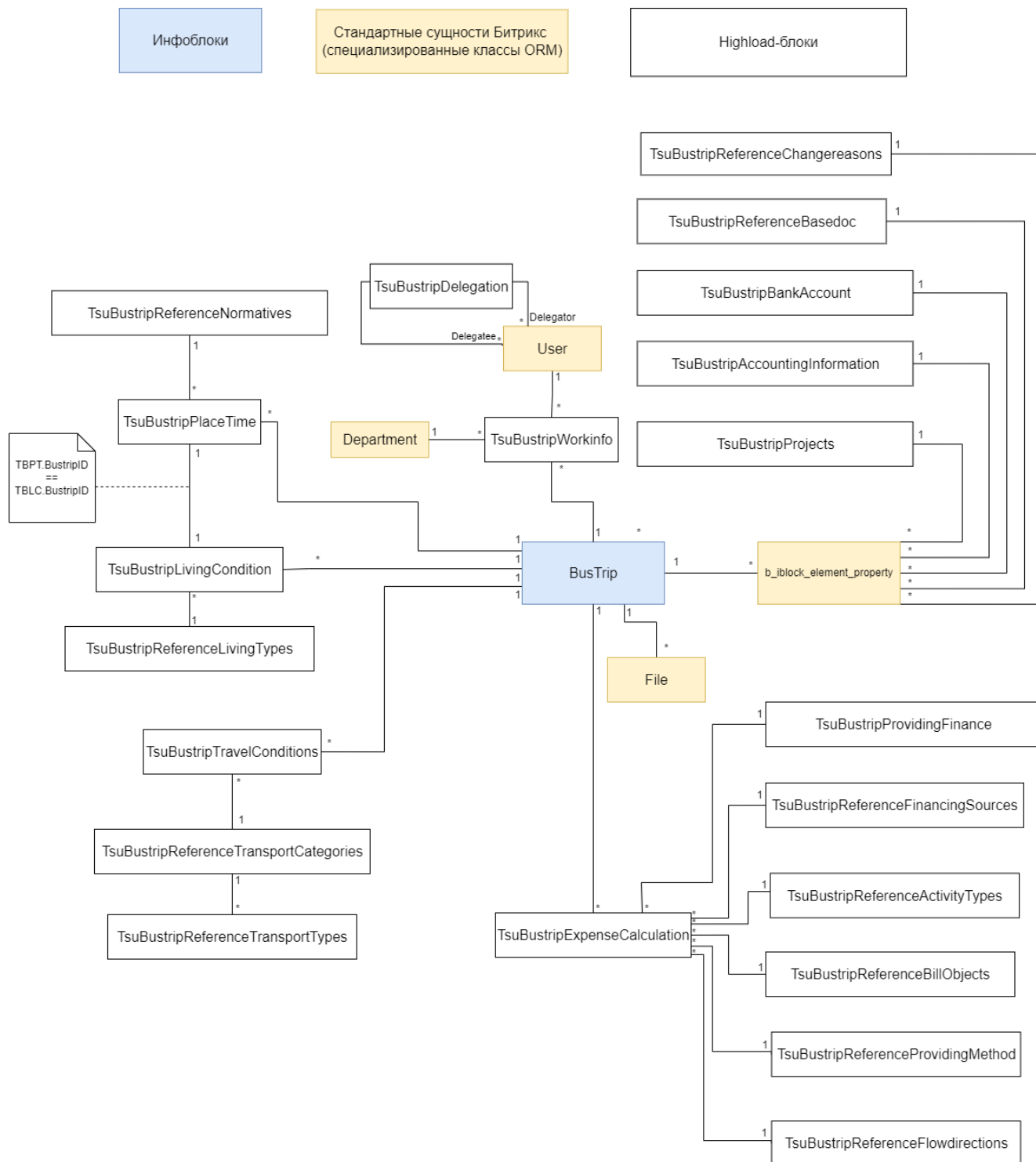


Рисунок 9 – схема реализованных таблиц для работы с командировками

Вспомогательные реализованные highload-блоки, не входящие в модель предметной области:

- TsuBustripWorkinfo

На ранних этапах разработки возникла проблема однозначного соотнесения должности сотрудника и его структурного подразделения, так как в базе пользователей отдельно хранились ID подразделений сотрудника и его должности в строковом виде. Чтобы пользователю не приходилось отдельно выбирать должность и подразделение в форме, и чтобы при этом избежать избыточной модификации стандартной пользовательской сущности, было принято решение создать отдельную таблицу с полями ID пользователя, ID подразделения и название должности. Данные по стон загружаются с persona.tsu.ru для 500 пользователей за раз. В форме создания командировки используется один селектор для элементов workinfo, отображающий одновременно должность и название подразделения по элементам.

После создания highload-блока было переделано регулярное обновление должностей и структурных подразделений сотрудников: теперь данные подгружаются в таблицу TsuBustripWorkinfo, а уже из него информация проставляется по пользователям сайта.

- TsuBustripDelegation

Требуется возможность создания заявки на командирование и прохождения процесса согласования не лично, а доверенным лицом человека, например, его секретарём. Соответственно, была создана таблица для хранения информации о делегированиях. Поля блока – ID делегирующего, ID пользователя, которому делегируют полномочия на заполнение формы, и сроки делегирования. Перед рендерингом формы создания командировки для текущего пользователя определяются все лица, которые делегировали ему полномочия на заполнение, чтобы определить доступные для выбора опции в селекторе «Командируемое лицо».

- TsuBustripAutocompleteFields

Ряд полей в форме создания командировок требует запуска каких-то событий на вводе или изменении значения. В highload-блоке TsuBustripAutocompleteFields записываются связанные поля и тип события, которое нужно выполнять при изменении одного из них. Далее эти данные подгружаются в script.js с помощью AJAX-действия и используются для создания событий.

- TsuBustripFieldHelp

Было сочтено необходимым оставить пояснения для определенных полей формы: по нажатии иконки пользователю показывается стандартный реализованный в JS-библиотеке Bitrix Framework alert, содержащий помощь по заполнению поля. В highload-блоке

указываются названия полей, для которых требуется отобразить такие иконки, и то, какой текст отображается при нажатии. Значения подгружаются в классе компонента `tsu.bustrip.create`.

Большинство остальных созданных таблиц представляют из себя справочники, ограничивающие выбор значений для конкретного поля.

Аналогичным образом была трансформирована модель предметной области для работы с авансовыми отчётами.

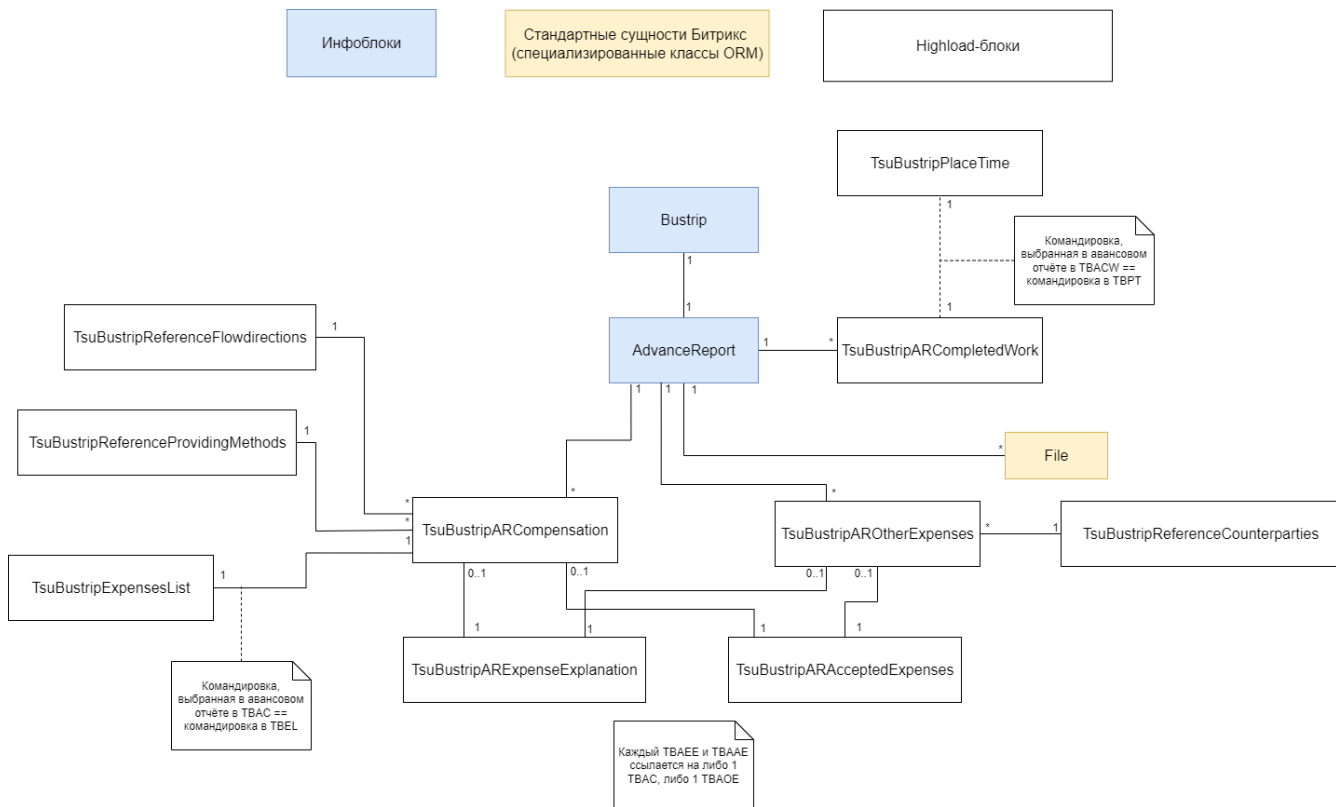


Рисунок 10 – схема реализованных таблиц для работы с авансовыми отчётами

Стоит обратить внимание на следующие моменты:

- Записи таблицы `TsuBustripARCompensation` («Компенсационные расходы по командировке») могут ссылаться на записи таблицы «Расчёт обязательств по командировке», если описывают расход, который был спланирован при заполнении связанного с отчётом решения о командировании. В таком случае ряд полей – количество и цена за единицу, сумма, наименование расхода, способ обеспечения – подставляются из связанного расхода по командировке.
- Записи таблиц `TsuBustripARExplanation` («Обоснование превышений нормативов») и `TsuBustripARAcceptedExpenses` («Принятые расходы»), несмотря на то

что в модели предметной области соответствующие сущности связывались напрямую с авансовым отчётом, здесь вместо этого содержат ссылку на конкретные расходы, так как описывают превышение нормативов и принятые суммы денежных средств по конкретным расходам. Каждая запись в этих двух таблицах может быть связана либо с одной записью в «Компенсационных расходах», либо с одной записью в «Прочих расходах».

- Содержимое таблицы `TsuBustripARCompletedWork` («Выполненная работа») по умолчанию генерируется автоматически в полном соответствии с содержимым таблицы `TsuBustripPlaceTime` («Места и сроки командирования») по связанной командировке.
- Записи по конкретному расходу из командировки в таблице `TsuBustripARExpenseExplanation` автоматически повторно генерируются при изменении информации по этому расходу в таблице `TsuBustripARCompensation` или `TsuBustripAROtherExpenses`.
- Аналогично командировке, для определения доступных для выбора значений в некоторых полях используется ряд справочников, таких как `TsuBustripReferenceFlowdirections` и `TsuBustripReferenceCounterparties`.

3.5.Компоненты

«Компонент – это логически завершённый код, предназначенный для извлечения информации из инфоблоков и других источников и преобразования её в HTML-код для отображения в виде фрагментов web-страниц. Состоит из собственно компонента (контроллер) и шаблона (представление). Компонент, с помощью API одного или нескольких модулей, манипулирует данными. Шаблон компонента выводит данные на страницу.» [5]

Компоненты должны находиться в папке `/bitrix/components` или `/local/components` реализовывать следующую базовую файловую структуру:

- `component.php` (или `class.php`; в дальнейшем будет упоминаться как `class.php` для краткости)
- `/templates`
 - `/ {name}`
 - `template1.php`
 - `template2.php`
 - `script.js`

- style.css

Класс, описанный в class.php, должен наследоваться от базового неабстрактного класса CBitrixComponent.

При вызове на странице метода includeComponent(componentName: string, templateName: string,arParams: array?) инстанса класса Application производится поиск одноименного компонента с указанным шаблоном в директориях /bitrix/components и /local/components. При нахождении создается объект класса, описанного в class.php, и в нём по порядку вызываются метод onPrepareComponentParams() (здесь рекомендуется первичная обработка переданных в компонент параметров) и затем метод executeComponent() (здесь располагается вся логика компонента по сбору и подготовке данных для рендеринга). При вызове в executeComponent() метода класса includeComponentTemplate(subTemplateName: string) создается объект CBitrixComponentTemplate, в котором выполняется код, описанный в выбранном финальном шаблоне. К шаблону автоматически подключаются скрипты из script.js и стили из style.css.

Для рендеринга шаблон может использовать данные, записанные в поляarResult иarParams компонента, обращаясь к ним как к параметрам функции (весь код, размещенный в файле шаблона, выполняется внутри основной функции класса шаблона). После отображения шаблона CBitrixComponent может использовать метод GetTemplate(), чтобы получить ссылку на объект CBitrixComponentTemplate. Шаблон может использовать метод GetComponent(), чтобы получить ссылку на объект CBitrixComponent.

Компоненты могут работать в режиме AJAX, для этого они должны реализовывать интерфейс Controllable. В этом случае компонент может определить ряд публичных функций, следующих паттерну наименования «*Action». Далее в произвольном javascript коде (при условии подключения js-библиотеки Битрикс) можно вызвать действие BX.Ajax.RunComponentAction, которому в качестве параметров передаются компонент, название действия («*Action»-именованной функции) и параметры для функции-действия. На контроллер /bitrix/services/main/ajax.php будет отправлен AJAX-запрос, в ходе выполнения которого будет подключён требуемый компонент и выполнено запрошенное действие. Метод возвращает Promise; в случае успеха в нём в формате JSON будет содержаться результат работы вызванного метода, в случае неудачи – выброшенные компонентом исключения. Компоненты, вызванные через этот метод, не выполняют методы onPrepareComponentParams() и executeComponent() перед запуском выбранного действия.

По умолчанию для авторизации таких запросов используется CSRF-токен, генерируемый на основе id сессии. Это позволяет предотвратить несанкционированный доступ к AJAX-действиям компонентов.

Основной повод использовать этот функционал – улучшение организации кода: таким образом возможно в одном месте держать весь код, напрямую связанный с работой компонента (чтобы из JS компонента вызывать только методы его класса), а также упрощение работы с AJAX-запросами в целом, благодаря готовой реализации роутинга, авторизации и перевода данных в JSON-формат.

В листинге 9 представлен пример действия.

Листинг 9 – действие, используемое для определения локации командировки при выборе вида шаблона edit компонента tsu.complex.hl.

```
public function getBustripLocationAction($element_id)
{
    $class =
    HLTable::compileEntity(HLTable::getById(Option::Get('tsu.bustrip',
    'IDiblockPlaceTime'))->fetch())->getDataClass();
    $foreignLocations =
    $class::getList(array('filter'=>array('UF_DOCUMENTID' => $element_id,
    '!=UF_COUNTRY'=>'Россия'))->fetchAll());
    if (count($foreignLocations) == 0)
        return 'РФ';
    else
        return 'Зарубежная';
}
```

В рамках реализации проекта были созданы три кастомных класса: tsu.bustrip.create, tsu.complex.hl и tsu.lists.list.

- tsu.bustrip.create – это компонент, используемый для рендеринга формы создания и редактирования заявки на командирование.

При первоначальном открытии ресурса /bustrip/create пользователь заполняет основную форму и нажимает на кнопку «Сохранить», после чего ему открываются на редактирование 4 таблицы с дополнительной информацией. По заполнении этих таблиц он нажимает на кнопку «Запустить», чтобы запустить бизнес-процесс согласования. Он может изменить основную информацию, ещё раз нажав на кнопку «Сохранить»; обновление информации также производится автоматически при нажатии на кнопку «Запустить».

В форме внимания заслуживают поля «Командируемое лицо», «Должность» и «Руководитель структурного подразделения», которые заполняются в связке и единообразно. Во время работы class.php вызывается метод setAvailableUsers(), который записывает в arResult ФИО и ID доступных для использования пользователей (список определяется с помощью класса BusTripDelegation как список из самого пользователя и всех других, которые делегировали ему право на заполнение командировки, при условии, что даты делегирования актуальны). В целях обеспечения единообразия дизайна формы для создания селекторов используется стандартный виджет Битрикс BX.UI.EntitySelector. На событие выбора предмета в селекторе командированного лица навешивается обработчик:

Листинг 10 – обработчик выбора командированного лица

```
'Item: onSelect': async function (event) {
    const dialog = event.getTarget();
    const data = event.getData();
    BX('USER_BT').value = data.item.id;
    updateOtherFields(BX('USER_BT').value);
    var positionDialog =
BX.UI.EntitySelector.Dialog.getById('dialog_WORK_POSITION_BT');
    positionDialog.removeItem();
    var workinfo = (await BX.ajax.runComponentAction('tsu:bustrip.create',
'getWorkinfoForSelector', {
        mode: 'class',
        data: {
            userId: BX('USER_BT').value
        }
    })).data;
    workinfo.available.forEach(element => positionDialog.addItem(element));
}
```

При выборе командированного лица все доступные на выбор элементы в селекторе «Должность» удаляются. После этого AJAX-метод BX.ajax.runComponentAction используется для того, чтобы в режиме AJAX обратиться к компоненту tsu.bustrip.create и использовать его метод для получения информации о доступных для пользователя соотношениях позиции и структурного подразделения, представленных в виде структуры данных элемента BX.UI.EntitySelector. Полученные элементы становятся доступными для выбора в селекторе «Должность».

Далее, при выборе или снятии выбора с элемента селектора «Должность» вызывается обработчик, который, во-первых, добавляет в форму скрытые input'ы позиции и структурного подразделения, и во-вторых, передает выбранные структурные подразделения в функцию updateDirectors(). После ещё одного AJAX-запроса функция получает данные о необходимых

руководителях структурных подразделений и на основании этих данных автоматически выбирает предметы в селекторе «Руководители структурных подразделений».

Для компонента `tsu.bustrip.create` было создано два шаблона – для командировки по общему и по научному направлению. Командировка по научному направлению содержит ряд дополнительных полей, связанных с выбором проекта. При выборе «Названия проекта» из селектора поля «Руководитель проекта», «Программа/Грант» и «№ Проекта» заполняются автоматически. Ввод значений в этих трех полях фильтрует список доступных для выбора названий проекта.

При нажатии на кнопку «Помощь» в верхней правой части основной формы открывается боковая панель с `iframe`, ведущим на запись в базе знаний (аналог вики) с инструкцией о заполнении командировки. По нажатию кнопки «История» доступен просмотр отфильтрованного для пользовательского просмотра лога бизнес-процесса в слайдере. Функции кнопок «Отмена» и «Корректировка» будут описаны ниже.

Рисунок 11 – основная форма командировки (компонент `tsu.bustrip.create`)

- `tsu.lists.list` – это компонент для просмотра командировок.

По большому счёту он реализован подобно стандартному компоненту `bitrix.lists.list`, но имеет следующие отличия:

- Поле «статус» выводится не текстом, а окрашенной в присвоенной статусу цвет ячейкой.
- При запуске компонента происходит проверка текущего пользователя на принадлежность определенным группам (группы пользователей – часть стандартной структуры данных). Если он относится к группе «Администраторы» или «Сотрудники отдела кадров», в списке выводятся все командировки. В противном случае выводятся командировки, принадлежащие этому пользователю и всем лицам, делегировавшим ему права на заполнение заявок за них.
- Гиперссылки на файлы открывают файл в `iframe` поверх страницы, а не в новой вкладке.

Название командировки	Статус	ID	Командируемое лицо	Мобильный телефон	Дата начала командировки	Структурное подразделение	Должность	Цель командировки
Зарубежная командировка №210	На исполнении	17449	(gala@mail.tsu.ru) Эдуард Галажинский	+73822529718	15.02.2022	Ректорат	Ректор	Обсуждение развития деятельности в образовательной сфере. Ректорском заседании «Программы управления университетом» качество деятельности «Управление консорциумом»
Зарубежная командировка №4	Заявка закрыта	16883	(dot_of_09@rambler.ru) Ольга Дюченко	+79138826485	04.12.2021	каф. радиотехники	Доцент	участие в 4 Future Skills Challenge в Финала V H Менеджеры чемпионата профессионалов (Ворлдскилл)
Командировка		16968	(solomita@mail.ru) Людмила Панкратова	+79039544697	13.12.2021	управление музейной деятельностью	Начальник управления	1. Участие в делегации Т. А. С. Пушкин 2. Знакомство с организацией фондовой и хранилища музейных (ТМ) Пушкина, М архивов. Музей архив А. Р. Пушкина

Рисунок 12 – список командировок (компонент `tsu.lists.list`)

- `tsu.complex.hl` – это компонент, используемый для рендеринга четырех заполняемых после основной формы таблиц, а также для добавления и редактирования элементов в них.

`tsu.complex.hl` использует два шаблона: `.default` и `.edit`. Шаблон `.default` используется для отображения грид-таблицы, в которой можно просматривать, редактировать, удалять и добавлять элементы. Шаблон `.edit` используется при редактировании или добавлении элемента

и представляет собой открываемую в слайдере форму, при отправлении которой происходит соответственно обновление существующего элемента или добавление нового.

Для упрощения управления гридом было также разработано инициализируемое в конце шаблона javascript-расширение `Tsu.HI`, методы которого используются для обработки нажатия на кнопки «Редактировать», «Удалить» и «Добавить», а также для обработки закрытия слайдера.

В листинге 11 представлена функция-обработчик нажатия на кнопку «Удалить». При нажатии на кнопку «Удалить» происходит вызов функции `HI.delete(elementId)`. Вызывается действие компонента (в данном случае – действие «delete»), по его результатам с помощью менеджера из JS-библиотеки находится текущий грид и полностью перезагружается.

Листинг 11 – обработчик удаления

```
delete(elementId) {
    var self = this;
    var request = BX.ajax.runComponentAction(self.componentName,
    self.actionDelete, {
        mode: 'class',
        data: {
            element_id: elementId,
            table: self.table
        }
    });
    request.then(function (response) {
        self.reloadGrid();
    });
    request.catch(function (response) {
        self.showErrors(response);
    })
}

reloadGrid() {
    var reloadParams = {apply_filter: 'Y', clear_nav: 'Y', table:
    this.grid_id};
    var gridObject = BX.Main.gridManager.getById(this.grid_id);

    if (gridObject.hasOwnProperty('instance')) {
        gridObject.instance.reloadTable('POST', reloadParams);
    }
    if (this.grid_id == "tsubustrip_bankaccount")
        location.reload();
}
```

Основная причина для кастомной реализации AJAX-действий с гридом – излишняя сложность и неконтролируемость стандартного взаимодействия с AJAX-компонентами. Теоретически, возможно было бы обойтись без этого js-расширения (если бы оно также не обрабатывало вывод ошибок при попытке создать/отредактировать элемент) и без переопределения стандартных функций кнопок грида. Однако в таком случае при

использовании действия «удаление» на компоненте со включённым режимом AJAX происходило бы в общих чертах следующее: [6]

1. Отправляется запрос на загрузку текущей страницы, к которому дополнительными параметрами добавляются ajax-id обновляемого компонента и параметры AJAX-действия.
2. Страница загружается обычным образом, пока не доходит до компонента под вопросом.
3. Буфер очищается, в качестве ответа отправляется актуальное содержимое компонента вместе с параметрами, которые дают обработчику ответа понять, что пришло AJAX-обновление.
4. Клиентский обработчик AJAX подставляет обновленные данные в нужный компонент.

Такие обновления сложнее модифицировать, чем если использовать специализированный метод для перезагрузки грид-таблицы, и их использование ударило бы по отзывчивости интерфейса (для каждого такого обновления данных требовалось бы сначала ожидать рендеринга страницы формы командировки, то есть исполнения практически всего кода в компоненте `tsu.bustrip.create` и 0-3 предшествующих `tsu.complex.hl`).

4 компонента `tsu.complex.hl` с шаблоном «.default» подключаются в конце шаблона `tsu.bustrip.create`, название используемого компонентом `highload`-блока передаётся в массиве `arParams`. Также компонент используется на странице настроек для добавления и просмотра банковских реквизитов. Для открытия компонента в шаблоне «.edit» создана отдельная страница, на которой компонент вызывается с нужным шаблоном; название `highload`-блока для открытия передаётся на страницу GET-параметром в открывающей страницу функции `openSidePanel()`.

Место и сроки командирования

Страна

Населенный пункт

Организация

Дата начала

Дата окончания

День выезда

Россия

г.Москва

МГУ

09.02.2022

10.02.2022

09.02.2022

ДОБАВИТЬ

Условия проезда

Транспорт

Пункт отправления

Дата отправления

Пункт прибытия

Дата прибытия

Категория

Воздушный

Томск

09.02.2022

Москва

09.02.2022

Эконом

ДОБАВИТЬ

Условия проживания

+

Расчёт обязательств по командировке

+

Рисунок 13 – компонент *tsu.complex.hl* в шаблоне *.default*

Условия проезда

Нужна помощь по заполнению?

Условия проезда

Страна отправления *

Россия x v

Транспорт *

Автомобильный [1] v

Пункт отправления *

Дата отправления *

Признак корректировки норматива

☐ да

Страна прибытия *

Россия x v

Категория (класс) проезда *

Автобус v

Пункт прибытия *

Дата прибытия *

* - поля обязательные для заполнения

СОХРАНИТЬ

ОТМЕНА

Рисунок 14 – компонент *tsu.complex.hl* в шаблоне *.edit*

Использование AJAX в компоненте выполняет три задачи.

1. Грид перезагружается, когда пользователь определяет отображаемые в шаблоне .default поля элементов.
2. Грид перезагружается, когда пользователь вносит любые изменения в сам грид – то есть удаляет элемент, добавляет новый или редактирует существующий.
3. Грид перезагружается, если он является связанным для грида, по которому только что произошли изменения.

Рассмотрим поподробнее последний пункт. Вследствие высокого уровня зависимости записей в более поздних таблицах от более ранних, изменение элементов более высоко расположенных таблиц часто требует изменения элементов таблиц ниже. Это принимает несколько форм, например:

- При работе с таблицей 2.1, «Расчёт обязательств по командировке», общее правило – что для каждого элемента в таблице «Места и сроки командирования» должен быть описан соответствующий расход типа «Суточные» и для каждого элемента в таблице «Условия проживания» должен быть описан расход типа «Проживание»; в противном случае, скорее всего, заявка будет отклонена при согласовании с бухгалтером в «ТГУ.Контрагенты» (так как пользователь скорее всего забудет описать все расходы самостоятельно);
- Похожая ситуация актуальна для таблицы 2.2, «Обоснование расходов, отличных от установленных нормативов». Для записей, описывающих командировку на территории РФ, возможно превышение стандартных нормативов проезда и проживания. В таком случае должны автоматически создаваться записи, описывающее происходящее превышение стандартного расхода;
- В таблице 1.2, «Условия проезда», доступные для выбора страны отправления и прибытия определяются перечисленными в таблице «Места и сроки командирования» странами-местами командирования, и при исчезновении страны из списка используемых соответствующим образом должны обновляться записи в условиях проезда;
- Расходы по суточным должны перерасчитываться в зависимости от того, как пересекаются даты начала и окончания в таблице «Места и сроки командирования»: если на одну и ту же дату приходятся две записи, одна из которых происходит на

той же территории, что и предыдущая запись, а другая – на территории новой страны, норматив за этот день должен рассчитываться для новой страны;

- При работе с авансовыми отчётами, записи в таблице «Обоснование расходов, отличных от установленных нормативов» должны генерироваться автоматически на основании записей в таблице 1.1 «Отчёт о расходах в командировке (по компенсационным расходам)». Каждая запись в таблице должна описывать положительное или отрицательное расхода по авансовому отчёту от расхода, указанного в командировке, поэтому с каждым изменением любой записи в таблице 1.1 все обоснования расходов по использованному в записи командировочному расходу полностью перерасчитываются.

Пункт 4 реализован с помощью AJAX-действия в компоненте `tsu.complex.hl`, которое при всех 3 типах изменения грида «Места и сроки командирования» дополнительно вызывается перед его обновлением. Остальные реализованы с помощью событий, зарегистрированных в модуле `tsu.bustrip` (таким образом необходимость их запуска проверяется только на страницах, на которых подключен модуль `tsu.bustrip`).

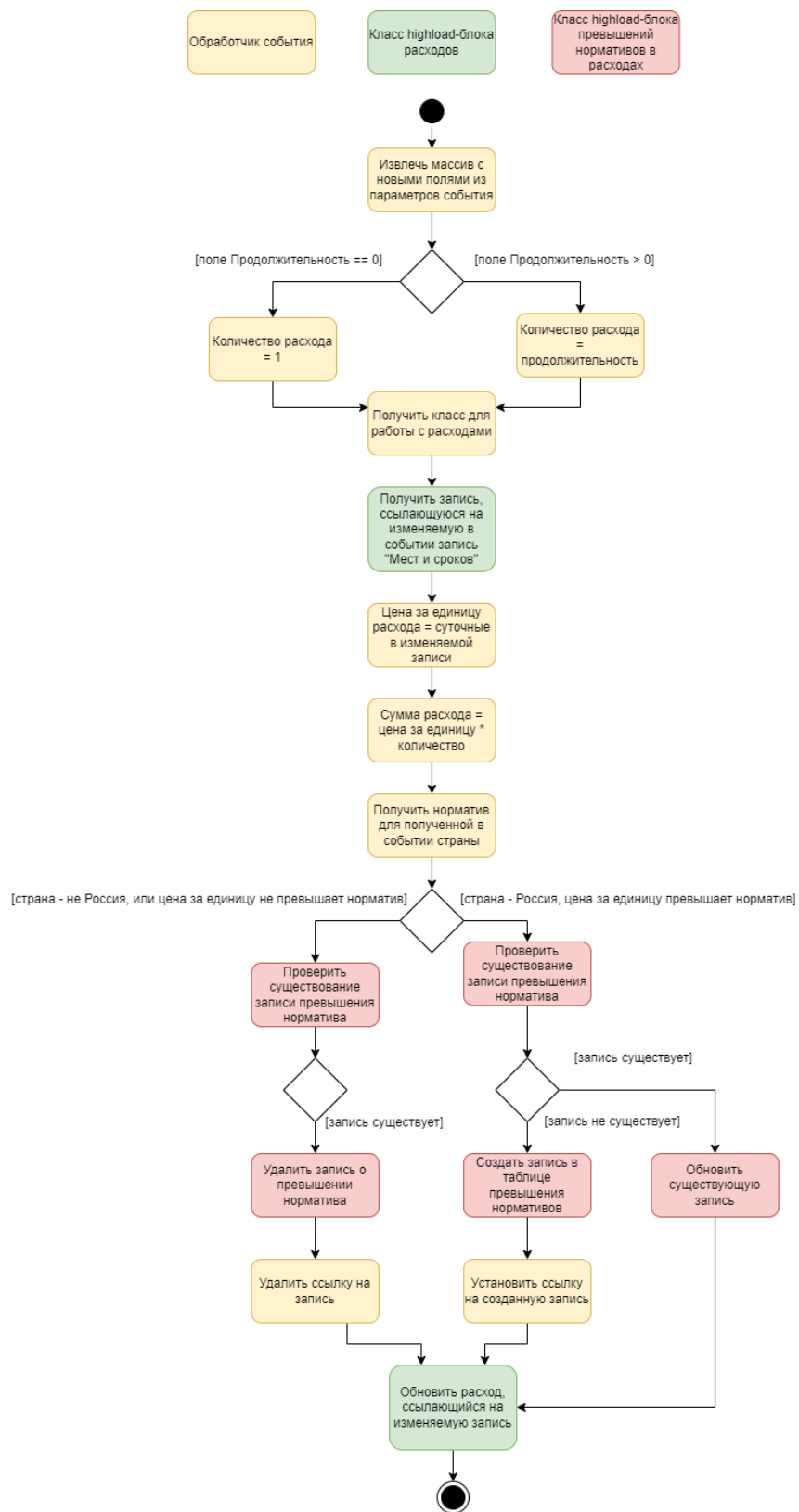


Рисунок 15 – событие, срабатывающее до обновления записи highload-блока *TsuBustripPlaceTime*.

3.6. Бизнес-процессы

«Бизнес-процесс (БП) – это процесс обработки документа, для которого задана одна точка входа и несколько точек выхода и последовательность действий (шагов, этапов, функций), совершаемых в заданном порядке и в определенных условиях. Другими словами, Бизнес-процесс – последовательность действий, направленных на получение заданного результата.» [7]

Процесс согласования заявки на командирование реализован с использованием модуля «Бизнес-процессы».

С точки зрения реализации бизнес-процесс можно разбить на две основные части: таблицы `b_bp_workflow_template` и `b_bp_workflow_instance`, и набор классов модуля.

Таблица `b_bp_workflow_template` содержит описания шаблонов бизнес-процессов. Таблица `b_bp_workflow_instance` содержит данные о конкретных экземплярах какого-то шаблона бизнес-процесса (бизнес-процессах).

Шаблон бизнес-процесса определяет его название, описание, то, с каким типом данных он работает (в нашем случае, например, это инфоблок с ID 63), а также его действия. Весь бизнес-процесс состоит из древовидной структуры действий, где тип самого бизнес-процесса – например, бизнес-процесс со статусами или последовательный бизнес-процесс – является корневым действием, а прочие действия могут выступать в роли родителей или потомков в зависимости от своего типа. Действие – это какая-то операция, изменяющая состояние документа, над которым выполняется бизнес-процесс, и/или самого бизнес-процесса; в том числе это может быть произвольный PHP-код. Действие одного и того же типа может вызываться с разными параметрами – например, действие «Приостановка процесса» может быть вызвано с произвольным временем ожидания. Работа классов действий будет описана позже; конкретно в поле `TEMPLATE` таблицы `b_bp_workflow_template` хранятся названия действий и то, с какими параметрами они будут вызываться в бизнес-процессе.

Основные классы, фигурирующие в работе бизнес-процессов, – это `CBPDocument`, `CBPRuntime`, `CBPWorkflow`, `CBPWorkflowPersister` и `CBPActivity`. При запуске метода `StartWorkflow()` класса `CBPDocument` получается инстанс `CBPRuntime` – класса исполняющей среды бизнес-процессов – и в ней с помощью метода `CreateWorkflow` создаётся объект управляющего выполнением действий БП класса `CBPWorkflow`, в котором инициализируются тип шаблона БП, документ, над которым будет производиться БП, и действия БП с используемыми параметрами. После этого `CBPDocument` запускает полученный экземпляр

CBPWorkflow, и тот по очереди запускает все описанные в БП до ближайшей остановки CBPActivity (действия). После выполнения всех доступных на данный момент действий CBPWorkflow использует класс CBPWorkflowPersister, чтобы сохранить себя в виде записи в таблице b_bp_workflow_instance. При выходе бизнес-процесса из ожидания – например, при получении rest-контроллером внешнего события, когда бизнес-процесс находится в требующем пользовательского участия действии (задании), – CBPWorkflowPersister используется, чтобы на основе данных из b_bp_workflow_instance заново создать экземпляр класса CBPWorkflow, и тот снова инициализирует и обрабатывает актуальную очередь действий, после чего снова записывается в БД.

Нужно помнить, что, хотя после достижения бизнес-процессом приостановки в результате действия «Приостановка» возобновление бизнес-процесса и выполнение всех последующих действий будет произведено системным демоном, запуск бизнес-процесса или его возобновление по внешнему событию – это полностью синхронные последовательности вызовов, возвращающие результат только в конце обработки всех действий в очереди CBPWorkflow. Соответственно, если пользователь запускает бизнес-процесс, выполнение всех первичных действий которого занимает тридцать секунд, все эти действия будут выполнены в пользовательской сессии и заставят сайт на тридцать секунд зависнуть для пользователя. Ресурсоемкие последовательности действий стоит начинать с действий, быстро заканчивающих для пользователя выполнение текущей очереди (например, с приостановки на 1 секунду). К сожалению, в официальном курсе по разработке Vitrix-процессов об этом не предупреждается.

Создание и редактирование шаблонов бизнес-процессов обеспечивается с помощью полностью функционального графического редактора. Теоретически возможна ручная правка дерева действий, однако эта информация хранится в виде запакованного архива в поле типа mediumblob, и никаких инструкций на по этой теме не предлагается (хотя официальный курс по бизнес-процессам предлагает пример того, как это дерево выглядит в текстовом виде [8]). Скриншот установки параметров для действия «Изменение документа» в редакторе был приведён на странице 20.

Бизнес-процесс позволяет определять и назначать собственные локальные (свои для каждого БП шаблона) или глобальные (одинаковые для всех БП шаблона) переменные.

Для реализации процесса согласования использовался тип бизнес-процесса «Бизнес-процесс со статусами». Статусы – это именованные блоки действий. Эксклюзивное для этого типа БП действие «Установить статус» позволяет прервать выполнение текущего статуса и

начать выполнение указанного. Это оптимальный тип процесса для поставленной задачи, так как заявка на командирование в процессе согласования может, в случае отклонения, несколько раз попадать на один и тот же этап.

На рисунке 14 в виде диаграммы состояний представлены реализованные статусы бизнес-процесса согласования. Из отличий от планируемого бизнес-процесса, описанного в части ЧЧ, выделяются, во-первых, специальные статусы для обработки первой и второй отправки в «ТГУ.Контрагенты», и во-вторых, отдельный статус для обработки отмены согласования по инициативе запустившего БП.

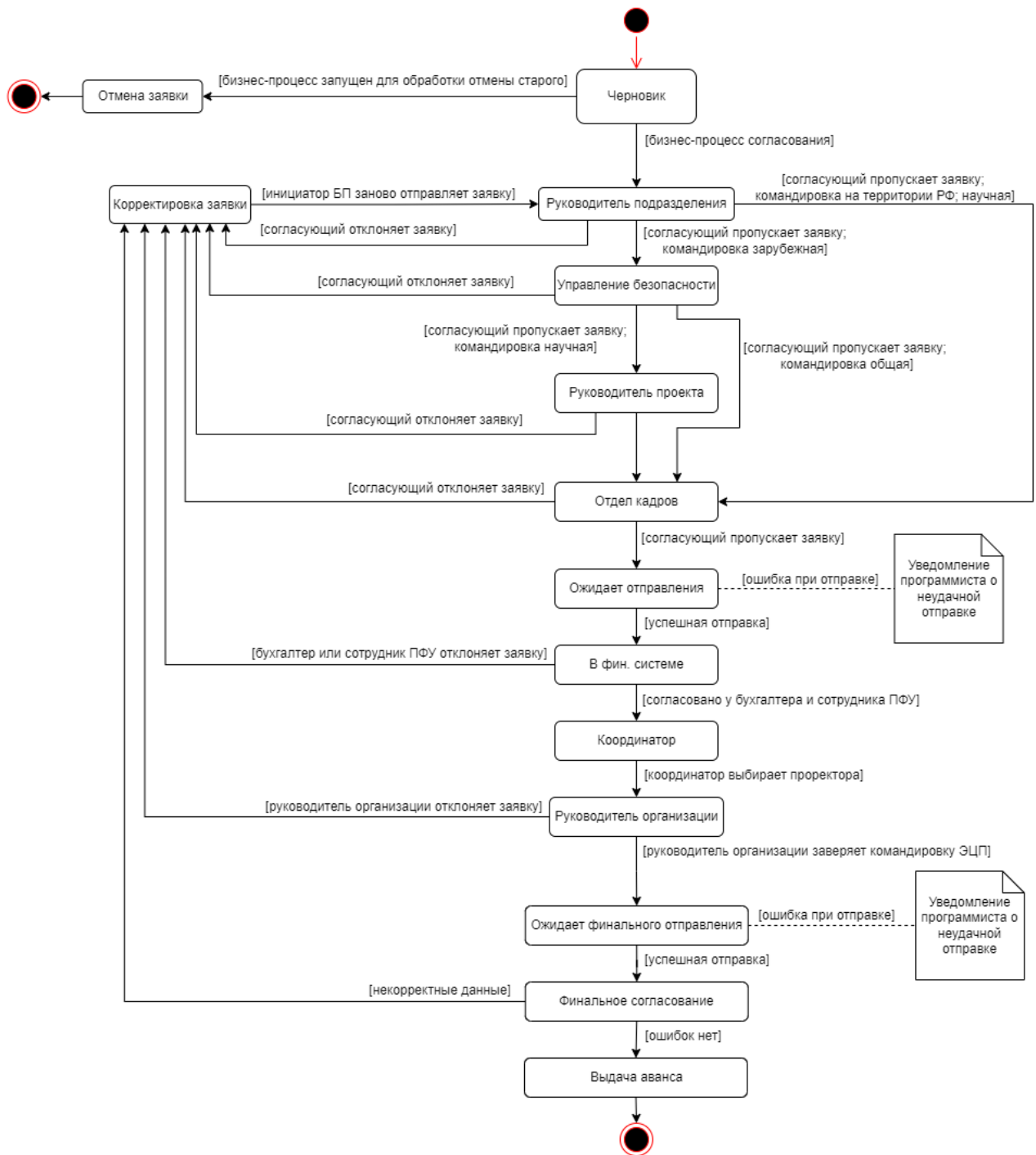


Рисунок 16 – статусы реализованного бизнес-процесса

Стандартные действия бизнес-процесса хранятся в папке /bitrix/activities/bitrix, но возможно создание кастомных действий в /bitrix/activities/custom. Основные файлы внутри

директории задания: {название действия}.php – класс действия, наследуемый от абстрактного класса CBPActivity; .description.php – описание результатов работы действия; properties_dialog.php – форма для установки параметров действия.

При разработке формы согласования первоначально планировалось определять однозначное соответствие руководителя подразделения и его должности по тому, какое структурное подразделение выбираются для командированного лица (для руководителя выбиралась должность, относящаяся к подразделению, руководителем которого являлся пользователь). Однако позже было решено дать заполняющему вручную выбирать руководителей структурного подразделения. Чтобы исключить возможность ошибки со стороны заполняющего форму — вместо того, чтобы заставлять его выбирать должность своего руководителя, было создано кастомное задание (действие, требующее пользовательского ввода для завершения) для согласования документа с выбором должности подписанта.

При достижении бизнес-процессом этого действия бизнес-процесс приостанавливается, указанным в параметре approve_users пользователям становится доступной для просмотра страница с описанием задания, указанным в параметре approve_description (просьба согласовать командировку, информация о командировке и документы). Перед созданием задания к этому описанию добавляется HTML-форма, описываемая в методе ShowTaskForm() класса действия, с селектором для выбора должности согласующего (из таблицы TsuBustripWorkinfo) и полем для комментария при отклонении, а также кнопками для согласования и отклонения. При вводе согласующим нужных данных в форме и нажатии на кнопку «Согласовать» или «Отклонить» отправляется rest-запрос, по итогам которого в классе действия выполняется метод OnExternalEvent. По выполнении этого метода в результаты выполнения задания записывается статус согласования (пропуск/отклонение), а также должность согласовавшего/комментарий.

Новое задание: укажите источник финансирования командировки №637/1 от 26.04.2022 16:13:34

Командировка №637/1

Задание:

1) Укажите источник финансирования командировки №637/1 от 26.04.2022 16:13:34 в специальном поле "Виза руководителя подразделения";
2) Нажмите кнопку "СОГЛАСОВАТЬ".

Командируемое лицо: Даниил Недосеков
Должность: Инженер II категории
Подразделение: центр компетенций по корпоративным информационным системам
Цель командировки: Тест наука 02 26.04.2022

Документы для согласования:
Недосеков Д.А. Изменение командировки №637_1.pdf

Служебная записка №17850 на разовый въезд_Шидловский.docx

Финансовая информация: Выполнение фундаментальных и прикладных научных исследований, проведение опытно-конструкторских, опытно-технических, опытно-технологических работ и производство перспективной техники и других изделий с учетом профиля подготовки кадров, Влияние глобальных и региональных природно-социальных изменений на выбросы и связывание углерода в почвах и водах от субарктики до экватора

Перейти к документу

* Виза руководителя подразделения:
укажите источник финансирования командировки

Комментарий при отклонении:

Выберите вашу должность*:

Стажер ▾

СОГЛАСОВАТЬ ОТКЛОНИТЬ ДЕЛЕГИРОВАТЬ

Рисунок 17 – кастомное задание «Ввод дополнительной информации с выбором должности»

После выполнения задания (когда оно получает ответы всех пользователей из approve_users и возвращает контроль CBPWorkflow) результаты его выполнения становятся доступными для использования в других действиях. Чтобы использовать должность согласовавшего в генерации документа, она сначала записывается в переменную бизнес-процесса (для приведения формата), после чего вызывается действие:

Заголовок: PHP код

PHP код:

```
Bitrix\Main\Loader::includeModule('tsu.bustrip');  
$functions = new Tsu\Bustrip\BusTripFormFunctions();  
$functions->addDirectorsSignatures({{ID элемента}}, $this->GetVariable('SelectedPositions')[0]);
```

Ид [Комментарий]

Рисунок 18 – Действие БП «PHP код», реализующее добавление подписи с информацией о должности подписавшего в поле «подписи» командировки.

Таким образом была решена проблема неоднозначности должности согласующего.

Другая возникшая в ходе разработки задача – проблема обработки бизнес-процессом изменений в документе. Например, после отклонения заявки одним из согласующих нужно дать пользователю возможность откорректировать заявку с использованием компонента

tsu.bustrip.create и затем отправить её на повторное согласование. Со стороны бизнес-процесса сброс согласования осуществляется просто: действие «Изменить статус» позволяет прервать выполнение текущего статуса БП и перейти к выполнению действий статуса «Корректировка», после чего ещё одно использование действие возвращает процесс в статус «Руководитель подразделения» для сброса согласования. Но корректировка проходит не мгновенно, и сам пользователь решает, когда она заканчивается, — как дать бизнес-процессу понять, что пользователь закончил изменение документа? Самым очевидным решением было бы отправить пользователю отдельное задание, которое он должен будет выполнить после корректировки, таким образом перезапуская процесс через внешнее событие; но это потребовало бы от пользователя отправки двух форм вместо одной. Другим вариантом могло бы быть внешнее изменение переменной БП со статусом корректировки, но хотя методы для использования переменных БП можно использовать вне его действий, после приостановки БП время, когда его переменные можно изменить без его полноценного возобновления, ограничено (без модифицирования данных в таблице b_bp_workflow_instance).

Соответственно, для решения проблемы было использовано действие БП «Цикл», в котором с помощью действия «Условие» проверяется значение поля документа «Статус корректировки». При обновлении пользователем заявки вызывается метод, который изменяет значение этого поля, и при следующей проверке цикл разрывается и выполнение БП продолжается. Внутри цикла используется действие «Приостановка процесса», чтобы проверять документ только 1 раз в минуту.

Аналогично реализованы циклы по попытке отправить заявку в «ТГУ.Контрагенты» и по ожиданию ответа оттуда (в данном случае для изменения полей документа используется stop-функция, осуществляющая взаимодействие с ресурсом, вместо пользовательского ввода).

Ещё одной проблемой было стандартное отображение файлов в заданиях. В действиях бизнес-процессов доступны для использования теги, через которые можно получить доступ к глобальным переменным, полям документа и переменным БП; при создании представления действия, например, конкретного оповещения или задания, эти теги заменяются на соответствующие значения. [9] При использовании тегов, соответствующих файловых полям документа, в представление действия подставляется название файла-гиперссылка на ссылку для его загрузки. Однако ссылка на загрузку даётся не с использованием абсолютного пути, а с маршрутизацией через компонент модуля «Списки» с использованием присваиваемого в рамках этого модуля ID файла, для проверки прав доступа пользователя к просмотру полей

конкретного элемента. В результате при обновлении файлового поля документа, например при регенерации решения о командировании, даже при сохранении абсолютного пути, «списочная» ссылка на файл меняется, и таким образом гиперссылка в задании становится недействительной. Для решения проблемы были введены функции, позволяющие получить в БП доступ к абсолютным файловым путям – подход был сочтён приемлемым ввиду того, что на сайте и так открыт неавторизованный доступ в папку /upload, где хранятся все нужные файлы.

Ещё один заслуживающий внимания момент разработки БП: в целом документация Bitrix рекомендует использовать набор JS-функций BX24.callMethod для работы с бизнес-процессами, например получения списка БП по элементу инфоблока, запуска БП по элементу или его удаления. Однако эти функции показали себя крайне ненадёжными. Метод bizproc.workflow.instance.list не всегда возвращает полный список запущенных бизнес-процессов (иногда выдавая 0 по элементу, по которому в данный момент активно 2 БП), а метод bizproc.workflow.kill, который первоначально планировалось использовать для удаления старого БП при отмене командировки, для большинства пользователей не работает в принципе: rest-контроллер проверяет права доступа пользователя к запущенному бизнес-процессу, чтобы понять, вернуть ему HTTP 403 Forbidden или вызвать метод удаления БП, но вызываемый для удаления БП метод в начале действия проверяет, является ли пользователь администратором, и всегда выбрасывает исключение в противном случае (возвращая клиенту HTTP 0 Unexpected Response). В результате для работы с БП напрямую используется класс CBPDocument, в том числе его незадокументированный [10] метод killWorkflow().

3.7. Процессы, поставленные на cron

Ряд функций, которые для нормальной работы проекта должны выполняться регулярно, поставлены на демон cron.

- Раз в минуту класс BusTripManager производит авторизацию в «ТГУ.Контрагенты», после чего сначала проверяет статус командировок, находящихся в статусе «В бухгалтерии», и соответствующим образом обрабатывает изменение их статуса в «ТГУ.Контрагенты», а затем производит отправку командировок, находящихся в

статусах «Ожидает отправления» и «Ожидает финального отправления». На рисунке 17 представлен процесс первичной отправки командировок.

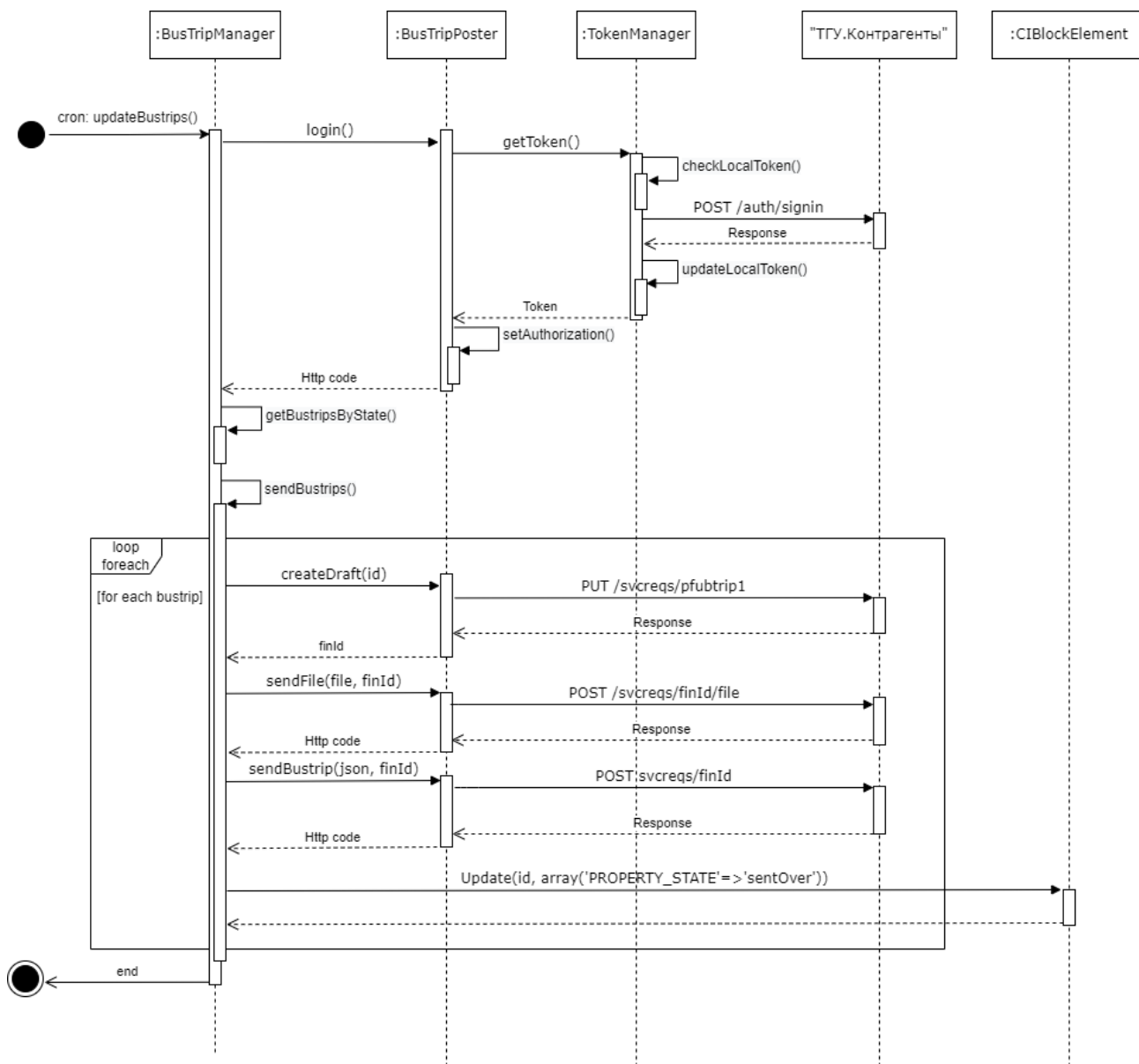


Рисунок 19 – отправка командировок в «ТГУ.Контрагенты»

Раз в час класс UserUpdater производит синхронизацию таблицы TsuBustripWorkinfo и полей «Должность» и «Структурное подразделение» в таблице пользователей с актуальными данными на сервисе «ТГУ.Сотрудники». Синхронизация производится для 500 сотрудников за раз (при текущем количестве пользователей, за сутки для всех сотрудников обновление производится дважды).

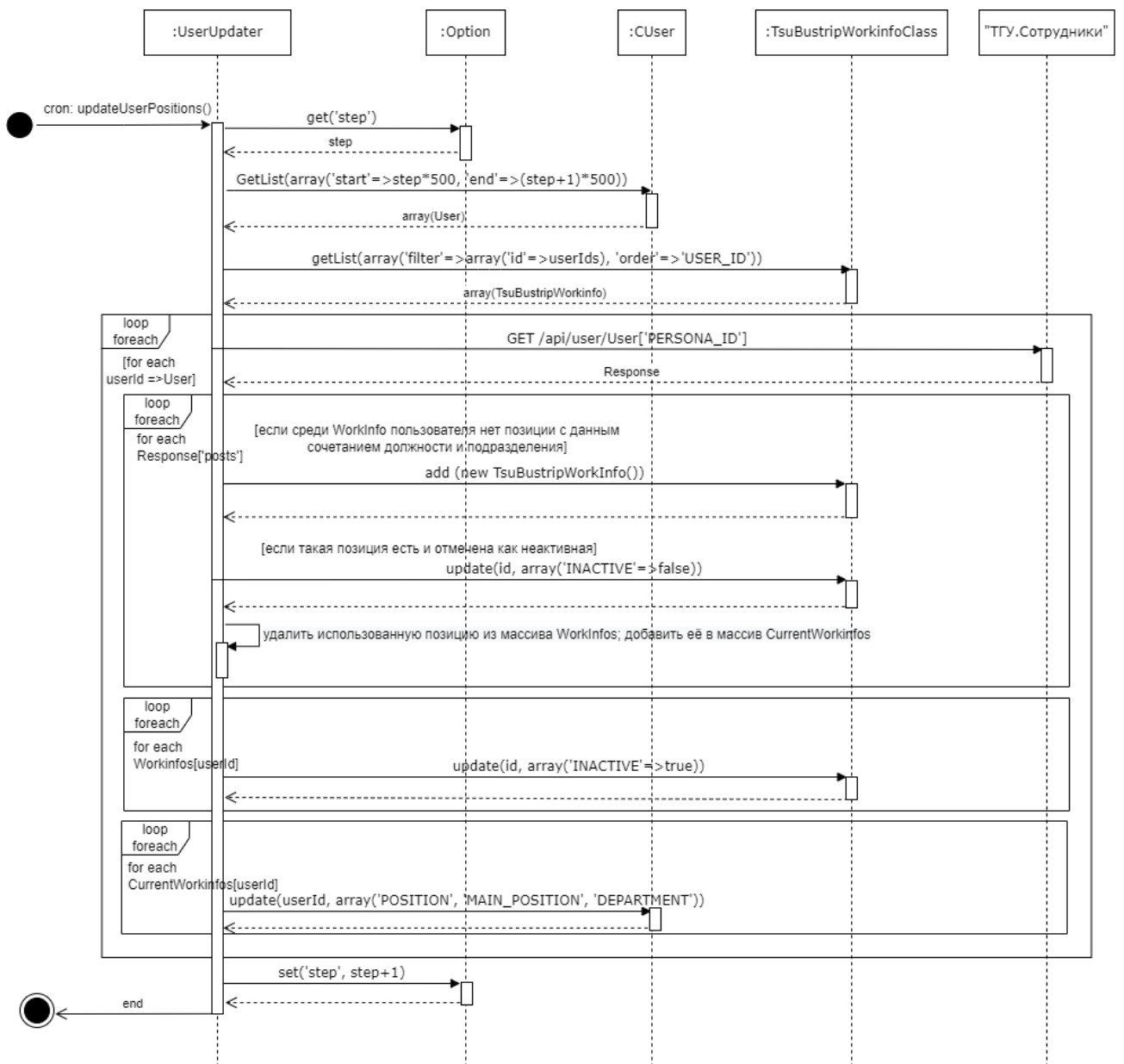


Рисунок 20 – обновление Workinfo и пользовательской информации

- Раз в сутки класс CurrencyUpdater производит синхронизацию используемых в нормативах для зарубежных стран курсов валют с данными на сайте Центрального банка РФ. Эти курсы валют используются для примерной оценки зарубежных нормативов в рублях на момент создания командировки.
- Раз в час класс BusTripNotifications используется, чтобы, в соответствии с хранимой в highload-блоке TsuBustripNotifications конфигурацией, оповещать соответствующих конкретных пользователей и соответствующие группы пользователей о командировках, застрявших в одном статусе больше чем на сутки (текст сообщения и оповещаемые пользователи определяются в зависимости от текущего статуса).

3.8.Генерация документов

В результате бизнес-процесса согласования генерируются следующие документы:

- документ решения о командировании или изменения решения о командировании в формате .docx;
- аналогичный документ в формате .pdf;
- версия документа для печати в формате .pdf.

Шаблон документа решения о командировании соответствует шаблону, утвержденному в приказе Минфина РФ от 30.03.2015 N 52н. [11]

Для генерации решений и изменений о командировании на основании предварительно созданных шаблонов в формате .docx используется библиотека PHPWord [12]. Библиотека позволяет считать .docx-файл и создать объект класса TemplateProcessor, реализующего набор функций для работы с виртуальным документом. По завершении работы функция saveAs() позволяет создать новый .docx-документ на основе виртуального.

Листинг 12 – создание виртуального документа

```
$this->processor = new TemplateProcessor($_SERVER['DOCUMENT_ROOT'] .  
"/upload/tsu-bustrip/".$this->BPID."/".$this->documentName");
```

Листинг 13 - замена тега {NAME} в документе на переданное значение

```
$this->templateProcessor->setValue("NAME", $user['NAME']);
```

Листинг 14 – создание виртуальной таблицы, её модификация и замена тега {PLACE_TIME} на созданную таблицу

```
$table = new Table(array('borderSize' => 8, 'layout'=>'fixed'));  
$row = $table->addRow();  
[...]  
$this->templateProcessor->setComplexBlock('PLACE_TIME', $table);
```

В ходе работы с библиотекой была обнаружена проблема: в текущей (последней, на момент создания класса) версии не работали функции для удаления и замены блоков (областей документа, обозначенных открывающим и закрывающим тегом). Для восстановления работы функции регулярное выражение было заменено:

Листинг 15 – регулярное выражение для поиска блоков

```
'/((<?xml.*)(<w:p\b.*>\${' . $blockname . '}</w:.*?p>)(.*)(<w:p\b.*\${' .  
$blockname . '}</w:.*?p>)/is'
```


Место и сроки командирования

Ответственный исполнитель	<u>{HRO_POS}</u>	<u>{HRO_BX}</u>	<u>{HRO_NAME}</u>	<u>{HRO_DATE_1}</u>
кадрового подразделения	(должность)	(подпись)	(расшифровка подписи)	

TRAVEL CONDITIONS
TRAVEL CONDITIONS DATA

{\LIVING_CONDITIONS}
 {\LIVING_CONDITIONS DATA}

φορμα 0504512 c.2

1. Условия командирования

1.1. Служебное задание на командирование

Место и сроки командирования

№ п/п	Место назначения		Срок командирования						Норматив (суточные), (руб)	Дополнительное условие
	населенный пункт	организация	дата начала	дата окончания	день выезда	день приезда	продолжительность (дн)	продолжительность с учетом дней в пути (дн)		
1	г. Москва	МГПОУ им. О.Е. Кутафина; Центр развития менеджмента, клиник	14.04.2022	17.04.2022	14.04.2022	18.04.2022	4	5	100	

Ответственный исполнитель кадрового подразделения	<u>Начальник отдела</u> (должность)	Подписано в Битрик24 (подпись)	<u>Вагнер Н.Г.</u> (подпись)	<u>05 апреля 2022 г.</u>
--	--	--------------------------------------	---------------------------------	--------------------------

1.2. Условия проезда

№ п/п	Вид транспорта	Маршрут				Категория (класс) проезда	Признак корректировки норматива
		отправление		прибытие			
		пункт	дата	пункт	дата		
1	Воздушный	Томск	14.04.2022	Москва	14.04.2022	Эконом	
2	Воздушный	Москва	17.04.2022	Томск	18.04.2022	Эконом	

1.3. Условия проживания

№ п/п	Условия проживания	Даты		Продолжительность (суток)	Категория проживания		Признак корректировки норматива
		заезда	выезда		содержание	сумма	
1	Гостиница	14.04.2022	17.04.2022	3	-	1650	12129

Для конвертирования .docx-документов в .pdf-формат используется LibreOffice. Так как LibreOffice не предоставляет PHP-библиотеки, работа с ней ведётся через командную строку:

```
$execCommand = "/bin/libreoffice7.1 --headless --convert-to pdf  
$path/$pdfName --outdir $path";  
exec($execCommand);
```

ЗАКЛЮЧЕНИЕ

В данной работе были описаны реализованные аспекты системы. Цели и задачи выполнены частично.

Часть системы, ответственная за работу с командировками, реализована и находится на стадии внедрения, с на данный момент 11 реализованными через неё командировками, однако перечисленные в параграфе 1.3 нефункциональные требования пока выполнены не полностью, в первую очередь из-за отсутствия решения по новой вёрстке. Ещё в январе 2022 г. были созданы макеты для страниц в соответствии с последними перечисленными нефункциональными требованиями, но на данный момент их использование уже не рассматривается, и неясно, когда требования по верстке будут утверждены.

Часть системы, ответственная за работу с авансовыми отчётами, находится на стадии разработки: только в середине мая 2022 г. были согласован актуальный вариант бизнес-процесса согласования для отчётов и окончательная структура данных для отправки в «ТГУ.Контрагенты». Основные усилия в разработке сейчас направлены на развитие этой части системы. На данный момент реализованы ввод информации инициатором и генерация первоначального документа.

Также рассматривается коммерциализация готового решения через его упаковку в устанавливаемый модуль Bitrix24 и его распространение через 1С-Битрикс Маркетплейс.

СПИСОК ЛИТЕРАТУРЫ

1. Архитектура MVC для Bitrix Framework // Архитектура продукта — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&LESSON_ID=2817 (дата обращения 14.03.2022)
2. Модули в Bitrix Framework // Модули — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&CHAPTER_ID=04609 (дата обращения 16.03.2022)
3. Инфоблоки в Bitrix Framework // Инфоблоки — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&LESSON_ID=4610 (дата обращения 18.03.2022)
4. SQL-представление свойств элемента инфоблока — [Б. м.], 2022 — URL: <https://bitrix.pushorigin.ru/orm/view-element-property> (дата обращения 25.04.2022)
5. Компоненты в Bitrix Framework // Что такое компонент — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&CHAPTER_ID=758 (дата обращения 23.03.2022)
6. AJAX в компонентах Bitrix Framework // Руководство по использованию технологии AJAX — [Б. м.], 2022 — URL: https://yunaliev.ru/file/bitrix_ajax_api.pdf (дата обращения 18.04.2022)
7. Введение в бизнес-процессы Bitrix Framework // Общая информация для разработчиков — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=57&LESSON_ID=23030&LESSON_PATH=5442.4567.23030 (дата обращения 27.03.2022)
8. Шаблон бизнес-процесса — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=57&LESSON_ID=3465&LESSON_PATH=5442.4567.3465 (дата обращения 05.04.2022)
9. Форма «Вставка значения» — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=57&CHAPTER_ID=05059&LESSON_PATH=5442.5446.5059 (дата обращения 11.04.2022)
10. Бизнес-процессы: класс CBPDocument — [Б. м.], 2022 — URL: https://dev.1c-bitrix.ru/api_help/bizproc/bizproc_classes/CBPDocument/index.php (дата обращения 05.04.2022)
11. Шаблон документа решения о командировании // Решение о командировании на территории Российской Федерации (ОКУД 0504512), от 30.03.2015 — URL:

http://www.consultant.ru/document/cons_doc_LAW_7445/1dd2334e0dc054f1be3f3b98c5a6357d6a7428f1/

12. Документация по PHPWord // Introduction: PHPWord 0.18.2 documentation — [Б. м.], 2022 — URL:

<https://phpword.readthedocs.io/en/latest/intro.html> (дата обращения 11.04.2022)

Отчет о проверке на заимствования №1



Автор: Трубников Владислав Игоревич
Проверяющий: Крестов Константин (Vladtrubn@gmail.com / ID: 8424381)
Отчет предоставлен сервисом «Антиплагиат» - <http://users.antiplagiat.ru>

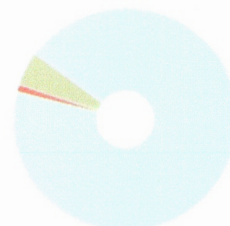
Зинев О.А.
10.06.2022

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 3
Начало загрузки: 08.06.2022 13:17:20
Длительность загрузки: 00:00:07
Имя исходного файла: Трубников В.И. ВКР.pdf
Название документа: Трубников В.И. ВКР
Размер текста: 1 кБ
Символов в тексте: 80036
Слов в тексте: 9376
Число предложений: 499

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 08.06.2022 13:17:28
Длительность проверки: 00:01:11
Комментарии: не указано
Поиск с учетом редактирования: да
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по eLIBRARY.RU (KkRu), Переводные заимствования по eLIBRARY.RU (KyRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования по Интернету (KkRu), Переводные заимствования по Интернету (KyRu), Переводные заимствования (KkEn), Переводные заимствования (KyEn), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, СПС ГАРАНТ, Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования



ЗАИМСТВОВАНИЯ
1,16%

САМОЦИТИРОВАНИЯ
0%

ЦИТИРОВАНИЯ
4,65%

ОРИГИНАЛЬНОСТЬ
94,19%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[01]	2,74%	не указано	13 Янв 2022	Библиография	
[02]	1,7%	не указано	13 Янв 2022	Цитирование	
[03]	0%	2015_ВКР_ПИЭ-104_КарелинаЕИ.docx	17 Июнь 2015	Кольцо вузов	
[04]	0%	ponomareva_a_s_avtomatizaciya-biznes-processov-vnutrennego-dokumentooborota-kompanii.pdf	31 Мая 2020	Кольцо вузов	
[05]	0,05%	Скачать - 1,4 МБ http://nauchkor.ru	22 Авг 2018	Интернет Плюс	
[06]	0%	Создание модулей для системы управления контентом Битрикс: выпускная квалификационная работа бакалавра http://bibliodub.ru	21 Янв 2020	Сводная коллекция ЭБС	
[07]	0%	Скачать - 1,4 МБ http://nauchkor.ru	22 Авг 2018	Интернет Плюс	
[08]	0%	5-ПИ_ВнуковаЕА_ВКР.docx	24 Июнь 2020	Кольцо вузов	
[09]	0%	https://nauchkor.ru/uploads/documents/5ab62fde7966e145c68c6977.pdf https://nauchkor.ru	24 Дек 2020	Интернет Плюс	
[10]	0%	https://nauchkor.ru/uploads/documents/5ab62fde7966e145c68c6977.pdf https://nauchkor.ru	07 Июнь 2019	Интернет Плюс	
[11]	0%	https://nauchkor.ru/uploads/documents/5ab62fde7966e145c68c6977.pdf https://nauchkor.ru	23 Мая 2022	Интернет Плюс	

1С-Битрикс: Управление сайтом Быстрый старт разработчика

[12]	0,05%	Содержание Введение Глава 1. Предварительные замечания ЧТО ТАКОЕ BITRIX FRAMEWORK КАК http://lib.knigi-x.ru	07 Ноя 2017	Интернет Плюс
[13]	0%	Скачать - 848,1 КБ http://nauchkor.ru	22 Авг 2018	Интернет Плюс
[14]	0%	Компоненты https://dev.1c-bitrix.ru	08 Июн 2022	Интернет Плюс
[15]	0%	https://saitsozdanie.ru/download/bitriks_q_start_developer_full.pdf https://saitsozdanie.ru	31 Янв 2022	Интернет Плюс
[16]	0%	https://saitsozdanie.ru/download/bitriks_q_start_developer_full.pdf https://saitsozdanie.ru	22 Мая 2022	Интернет Плюс
[17]	0%	Диплом Мелкумян	10 Июн 2016	Кольцо вузов
[18]	0%	Артамонов	23 Мар 2016	Кольцо вузов
[19]	0,04%	ОНТОЛОГИЧЕСКИЙ ПОДХОД ПРИ СИСТЕМАТИЗАЦИИ ВХОДНЫХ ДАННЫХ НЕЙРОННЫХ СЕТЕЙ. http://elibrary.ru	30 Авг 2017	Перефразирования по eLIBRARY.RU
[20]	0,03%	Применение технологий нейронных сетей для обработки данных в системе управления содержимым. http://elibrary.ru	15 Янв 2017	Перефразирования по eLIBRARY.RU
[21]	0%	Скачать Часть 2 (pdf) (4/9) http://moluch.ru	07 Янв 2017	Перефразирования по Интернету
[22]	0%	Применение технологий нейронных сетей для обработки данных в системе управления содержимым Публикация в журнале «Молодой ученый» http://moluch.ru	28 Янв 2017	Перефразирования по Интернету
[23]	0%	https://elar.rsvpu.ru/bitstream/123456789/12333/1/RSVPU_2016_096.pdf https://elar.rsvpu.ru	11 Апр 2022	Интернет Плюс
[24]	0%	Пример использования действия "Выбор сотрудника" https://dev.1c-bitrix.ru	08 Июн 2022	Интернет Плюс
[25]	0%	Большое внедрение и настройка CRM Битрикс. Внешняя интеграция https://intervolga.ru	06 Мар 2022	СМИ России и СНГ
[26]	0%	Шемонаева Светлана магистерская Шемонаева.docx	18 Янв 2018	Кольцо вузов
[27]	0,08%	Бизнес-процессы https://dev.1c-bitrix.ru	08 Июн 2022	Интернет Плюс
[28]	0%	Бизнес-процессы https://dev.1c-bitrix.ru	08 Июн 2022	Интернет Плюс
[29]	0%	Бизнес-процессы https://dev.1c-bitrix.ru	08 Июн 2022	Интернет Плюс
[30]	0%	ОНТОЛОГИЧЕСКИЙ ПОДХОД ПРИ СИСТЕМАТИЗАЦИИ ВХОДНЫХ ДАННЫХ НЕЙРОННЫХ СЕТЕЙ. http://elibrary.ru	30 Авг 2017	eLIBRARY.RU
[31]	0%	Применение технологий нейронных сетей для обработки данных в системе управления содержимым. http://elibrary.ru	15 Янв 2017	eLIBRARY.RU
[32]	0%	Курс для хостеров https://dev.1c-bitrix.ru	24 Янв 2021	Интернет Плюс
[33]	0%	Модули https://dev.1c-bitrix.ru	13 Ноя 2019	Интернет Плюс
[34]	0%	Настройка модуля https://dev.1c-bitrix.ru	19 Мар 2022	Интернет Плюс
[35]	0,12%	https://dev.1c-bitrix.ru/docs/chm_files/%D0%90%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80.%20%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B8.pdf (1/2) https://dev.1c-bitrix.ru	13 Ноя 2019	Интернет Плюс
[36]	0%	Разработка интернет-портала для ООО "УМ"	09 Июн 2011	Кольцо вузов
[37]	0%	диплом Уварова М.Н.	30 Мая 2012	Кольцо вузов
[38]	0%	Тунгатаров v1.doc	11 Мая 2014	Кольцо вузов
[39]	0%	ИСБУ_2014_ВКР(ДР)_Куртов И.А..docx	17 Мар 2014	Кольцо вузов
[40]	0%	2 1С-Битрикс: Управление сайтом Разработчик Bitrix Framework Содержание Введение 8 Глава 1. Что такое Bitrix Framework? КАК ИЗУЧАТЬ BITRIX FRAMEWORK? Как правильно построить http://lib.knigi-x.ru	07 Ноя 2017	Интернет Плюс
[41]	0,04%	ЗАКОНОМЕРНОСТИ И ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ В СОВРЕМЕННОМ ОБЩЕСТВЕ. http://elibrary.ru	раньше 2011	Перефразирования по eLIBRARY.RU
[42]	0%	ЗАКОНОМЕРНОСТИ И ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ В СОВРЕМЕННОМ ОБЩЕСТВЕ. http://elibrary.ru	раньше 2011	eLIBRARY.RU
[43]	0%	https://vkr.pspu.ru/uploads/729/Lopatin_vkr.pdf https://vkr.pspu.ru	13 Мар 2022	Интернет Плюс

[44]	0%	Типы операционных процессов, включая типы процессов в сервисных операциях. Измерение производительности операций. — контрольная работа http://turboreferat.ru	20 Мая 2020	Интернет Плюс	
[45]	0%	https://its.bsuir.by/m/12_130111_1_147692.pdf https://its.bsuir.by	22 Апр 2022	Интернет Плюс	
[46]	0%	1С-Битрикс: Корпоративный портал. Повышение эффективности компании. 2-е изд. http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	
[47]	0%	1С-Битрикс: Корпоративный портал. Повышение эффективности компании http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	
[48]	0%	РАЗРАБОТКА WEB-СИСТЕМЫ НА ОСНОВЕ ГОТОВОГО РЕШЕНИЯ В «1С-БИТРИКС:УПРАВЛЕНИЕ САЙТОМ». http://elibrary.ru	05 Авг 2016	Перефразирования по eLIBRARY.RU	
[49]	0,27%	РАЗРАБОТКА WEB-САЙТА НА ПЛАТФОРМЕ BITRIX DEVELOPMENT OF THE WEB-SITE ON THE BITRIX PLATFORM. http://elibrary.ru	17 Дек 2019	Перефразирования по eLIBRARY.RU	
[50]	0,23%	https://www.tsu.ru/upload/medialibrary/759/rekomenduemyy_shablon_programmy_gia-26.01.2021.docx https://tsu.ru	24 Янв 2022	Интернет Плюс	
[51]	0%	Глава 1. Общие моменты http://mybiblioteka.su	10 Янв 2017	Интернет Плюс	
[52]	0%	Задание параметров шаблона 19 Статусы 22 - Руководство - стр.1 http://gigabaza.ru	15 Апр 2017	Интернет Плюс	
[53]	0%	Стратегия развития бизнес процессов - презентация, доклад, проект https://myslide.ru	08 Июн 2022	Интернет Плюс	
[54]	0,23%	Всё о кешировании в Битрикс: D7, тегированный кеш, кеш в компонентах. - Роман Морозов https://r-morozov.ru	08 Июн 2022	Интернет Плюс	
[55]	0,21%	не указано	13 Янв 2022	Шаблонные фразы	
[56]	0%	https://lib.tsu.ru/win/produkcija/metodichka/pril1.pdf https://lib.tsu.ru	25 Мая 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[57]	0%	Что такое Bitrix / Хабр https://habr.com	08 Июн 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[58]	0%	Что такое Bitrix http://pcnews.ru	03 Янв 2019	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[59]	0%	Вагнер, Дмитрий Викторович Высокочастотные электромагнитные характеристики композиционных радиоматериалов на основе гексагональных ферритов : диссертация ... кандидата технических наук : 01.04.03 Томск 2019 http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[60]	0%	Ассаф Басем Разработка стратегии антикризисного управления гостиничными предприятиями : диссертация ... кандидата экономических наук : 08.00.05 Москва 2020 http://dlib.rsl.ru	16 Июн 2021	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[61]	0%	История медицины. Книга первая. Руководство к преподаванию http://studentlibrary.ru	26 Янв 2018	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[62]	0%	Организация профилактики заболеваемости новорожденных у беременных с туберкулезом и сопутствующими гепатитами В и С http://emil.ru	21 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[63]	0%	Административно-правовое обеспечение безопасности на железнодорожном транспорте http://dep.nlb.by	04 Июл 2017	Диссертации НББ	Источник исключен. Причина: Маленький процент пересечения.
[64]	0%	Интеграция Business Studio с ECM-системой DIRECTUM, Методика Контент-платформа Pandia.ru https://pandia.org	08 Июн 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[65]	0%	http://www.betec.ru/bstudio/bs-modeling-system.pdf http://betec.ru	08 Июн 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[66]	0%	Москва - главный донор России, запрет звуковой рекламы и есть ли польза от хламохранилищ https://oko-planet.su	02 Сен 2021	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[67]	0%	https://nmk.tusur.ru/storage/150764/conference-2022_1_new.pdf https://nmk.tusur.ru	24 Мая 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[68]	0%	https://rospatent.gov.ru/content/uploadfiles/tezisi-dokladov-24mkr.pdf https://rospatent.gov.ru	06 Апр 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[69]	0%	https://it-education.ru/conf2021/thesis/Thesis_IT_in_RF_2021.pdf https://it-education.ru	14 Мая 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[70]	0%	АНАЛИЗ УРОВНЯ ЖИЗНИ НАСЕЛЕНИЯ КЕМЕРОВСКОЙ ОБЛАСТИ - КУЗБАССА В КОНТЕКСТЕ РЕСУРСНОЙ ЗАВИСИМОСТИ ЭКОНОМИКИ РЕГИОНА – тема научной статьи по социологическим наукам читайте бесплатно текст научно-исследовательской работы в электронной библиотеке КиберЛенинка https://cyberleninka.ru	18 Апр 2022	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.