

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**МЕТОДЫ ИНТЕГРАЦИИ СПЕЦИАЛИЗИРОВАННЫХ  
ИНФОРМАЦИОННЫХ СИСТЕМ**

Магистерская диссертация  
обучающегося по направлению подготовки  
09.04.02 Информационные системы и технологии  
очной формы обучения  
группы 07001635  
Дегтяревой Анастасии Михайловны

Научный руководитель  
доцент, Гахов Р. П.

Рецензент  
к.т.н., доцент, Зайцева Т. В.

БЕЛГОРОД 2018

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. Анализ существующих подходов для интеграции информационных систем .....	6
1.1 Интеграция на уровне данных .....	6
1.2 Интеграция с помощью корпоративных приложений .....	7
1.3 Интеграция с помощью web-служб и web-API.....	8
2. Проектирование интеграции информационных систем .....	10
2.1 Информационная модель интеграции систем.....	10
2.2 Проектирование разделов интеграции систем .....	15
2.2.1 Проектирование раздела «Заказы питания».....	16
2.2.2 Проектирование раздела «Журнал взаимодействия».....	19
2.3 Проектирование интеграции информационных системам «Парус 8» и «ИСОУ «Виртуальная школа» с помощью веб-сервиса по средствам API....	21
2.3.1 Загрузка данных в раздел «Заказы питания» .....	21
2.3.2 Загрузка данных по учащимся.....	23
2.3.3 Выгрузка квитанций по лицевым счетам .....	26
2.3.4 Выгрузка данных по состоянию баланса лицевого счета .....	29
3. Программная реализация интеграции информационных систем .....	31
3.1 Разработка в ПП «Парус 8».....	31
3.2 Разработка web-API.....	41
ЗАКЛЮЧЕНИЕ .....	44
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	45
ПРИЛОЖЕНИЯ.....	48

## ВВЕДЕНИЕ

В настоящий момент взаимодействие информационных систем друг с другом является основой для существования и успешного развития компаний. Для пользователей интеграция систем решает проблемы, связанные с приемом и передачей данных, со временем обработки информации, с доступностью к данным и др. Информационные системы стали неотъемлемой частью современной экономики. К интеграции систем относится сбор различных подсистем в единое целое с физической или функциональной точки зрения. Глобализация вопроса о передаче данных привела к вынужденному обмену информацией между системами.

Научная новизна заключается в модернизации метода интеграции специализированных информационных систем «Виртуальная школа» и программного продукта (ПП) «Парус 8».

Объектом исследования выбран процесс учета оплаты питания в образовательных учреждениях. Предметом исследования являются методы и средства интеграции информационных систем.

Цель магистерской диссертации – совершенствование учета оплаты школьного питания за счет его автоматизации путем интеграции специализированной информационной системы образовательных услуг (ИСОУ) «Виртуальная школа» и программного продукта (ПП) «Парус 8».

К задачам магистерской диссертации относятся:

- анализ методов интеграции информационных систем;
- проектирование интеграции специализированных информационных систем;
- разработка интеграции специализированных информационных систем.

На сегодняшний момент существуют различные методы интеграции информационных систем. Для начала необходимо рассмотреть факторы, влияющие на интеграцию:

Безопасность – если данные передаются не из рук в руки, а передаются по каналам связи, вопрос о шифровании данных становится более актуальным.

Мобильность – пользователям необходимо реагировать и передавать данные мгновенно, за короткий промежуток времени, пользователь должен принять информацию, обработать и отправить ответ.

Непрерывность цикла работы – синхронизация и обновление систем не должно влиять на работу пользователей и функционирование системы в целом.

Высокая загруженность – количество пользователей, работающих в системе одновременно, поток обрабатываемой и передаваемой информации.

Интерактивность – пользователь системы всегда ожидает от системы большей скорости реагирования, быстродействия и оперативности обработки данных.

Межсистемная интеграция – взаимосвязь между системами партнеров, клиентов, поставщиков и т.д .

При решении задачи, связанной с межсистемной интеграцией сложность, заключается в следующих параметрах:

Концептуальные различия систем. Разработчики систем на этапе проектирования изначально использовали разные решения.

Технологические различия систем. Использование разных форматов данных, связей взаимодействия и сервисов.

Для решения сложных вопросов интеграции необходимо использовать следующий набор средств:

Стандартизация – использовать международные, государственные, отраслевые стандарты разработки.

Интеграция на уровне брокеров. Выбор этого средства имеет преимущество в том, что можно разработать дополнительный блок, к которому могут обращаться все системы и разными способами, например,

через базу данных и RPC. Недостатком такого средства является сложность и трудоемкость разработки.

Интеграция на уровне данных – возможность, при которой системы могут обращаться в одну базу данных. Преимущества такого средства заключаются в низкой стоимости интеграции. К недостаткам следует отнести тот факт, что если изменится структура базы данных, необходимо будет переписывать программный код всех процедур, приложений и отчетов.

Интеграция с помощью сервисов – современный и быстро совершенствующий подход интеграции систем. К преимуществам можно отнести взаимодействие интерфейсов и форматов данных, что способствует быстрой передаче данных.

При разработке интеграции информационной системы образовательных услуг (ИСОУ) «Виртуальная школа» и программного продукта (ПП) «Парус Бюджет 8» используется интеграция при помощи web-API. Управление информацией сервиса полностью основывается на протоколе передачи данных HTTP.

Магистерская диссертация выполнена на 48 листах, включает в себя 3 раздела, введение, заключение, список используемых источников и приложения.

## 1 Анализ существующих подходов для интеграции информационных систем

Интеграция – это взаимодействие разных систем, их блоков, передача данных в разных форматах. Интеграция может проходить на уровне данных, сетевых устройств, программных приложений, пользовательских интерфейсов и различных сервисов [1].

Рассматривая вопрос о межсистемной интеграции самой главной задачей становится найти оптимальный способ интеграции. Чистая система интеграции – трудоемкий и дорогостоящий процесс. Затраты на такую интеграцию увеличиваются за счет сопровождения чистой системы. Поэтому, для выбора оптимального способа интеграции необходимо рассмотреть возможные варианты и оценить их преимущества и недостатки.

### 1.1 Интеграция на уровне данных

При интеграции на уровне данных возникает много вопросов по типам и форматам данных. Преобразование разнovidной информации доставляет много проблем со сбором данных, их структурированием, обработкой, хранением и передачей [2].

Для интеграции на уровне данных, как правило используют стандартные протоколы и интерфейсы, такие как SQL.

Преимущества интеграции на уровне данных заключаются в низкой стоимости интеграции. К недостаткам следует отнести тот факт, что если изменится структура базы данных, необходимо будет переписывать программный код всех процедур, приложений и отчетов.

Самым простым примером, где используется такой вид интеграции – системы электронного документооборота, CRM и ERP-системы. Такие системы используют, изменяют информацию друг друга, как раз за счет интеграции данных.

## 1.2 Интеграция с помощью корпоративных приложений

Интеграция корпоративных приложений (EAI) это использование технологий и функций по всему предприятию, возможность интеграции программных приложений и аппаратных систем. Многие несвободные и открытые проекты обеспечивают поддержку решения EAI [3].

Развивающиеся связующие технологии EAI включают веб-службы интеграции, сервис ориентированные архитектуры, интеграции контента и бизнес-процессов.

Взаимосвязь между приложениями предприятия (EA), такие как управления отношениями с клиентами (CRM), управление цепочками поставок (SCM) и бизнес-аналитики не автоматизированы. Таким образом EAs не разделяют общие данные или бизнес-правила. EAI позволяет упростить и автоматизировать бизнес-процессы без применения чрезмерных изменений структуры приложения или данных [4].

При EAI возникают проблемы с несовместимостью операционных систем, архитектур баз данных и/или языков программирования, а также другие ситуации, где унаследованных систем больше не поддерживаются производителем.

Эти проблемы возникают, если следующие вопросы решаются с помощью EAI:

Интеграция данных - обеспечивает последовательную информацию между различными системами;

Независимость поставщиков - бизнес-политики или правила, касающиеся конкретные бизнес-приложения не должны осуществляться заново при замене части приложений.

Использование EAI при интеграции систем и/или приложений — это создание «брокера» информации, блок, к которому могут обращаться все системы и разными способами, например, через базу данных и RPC. Недостатком такого средства является сложность и трудоемкость разработки.

### 1.3 Интеграция с помощью web-служб и web-API

Для начала необходимо определиться чем web-API отличается от других web-служб. Все веб-службы являются API-интерфейсами, но не все API-интерфейсы являются веб-службами. Web-API и веб-службы часто путаются друг с другом, однако web-API - это эволюция веб-сервисов. Оба облегчают передачу информации, но web-API более динамичны, чем веб-сервисы [9].

По определению, веб-служба представляет собой программное обеспечение, которое доступно с помощью браузера. Клиент вызывает веб-службу, отправляя запрос, как правило в форме XML-сообщения, и служба отправляет ответ, также в формате XML. Связь между клиентом и веб-службой обычно создается с помощью HTTP.

Типичный веб-API определяет, как программные компоненты должны взаимодействовать друг с другом с использованием протокола HTTP. Клиенту не нужно знать, какую процедуру вызывать на сервере. Вместо этого он использует набор команд, которые встроены в HTTP, и когда команда поступает с другого конца, система получает информацию о том, что с ней делать [11]. Главным преимуществом веб-API является гибкость. Клиентская система и обслуживающая система («поставщик») настолько независимы друг от друга, что каждый из них может использовать разные языки (Java, Python, Ruby и т. Д.). Для своей части общая реализация. Кроме того, полезная нагрузка данных может быть нескольких типов, таких как JSON или XML. API-интерфейсы RESTful чаще всего используют протокол HTTP.

API определяет, как компоненты разных систем должны взаимодействовать друг с другом. API представляет собой набор протоколов и подпрограмм; ответы, как правило, возвращаются как данные JSON или XML. API могут использовать какой-либо протокол связи и не ограничиваться только им. тем же способом, что и веб-служба.



Веб-API и веб-службы служат средством интеграции систем. Оба поддерживают данные на основе XML, но JSON – это более распространенный текстовый формат обмена данными для веб-API. При сравнении веб-сервисов с веб-API значимость заключается в объеме работы, которую должны обработать сервисы. Процессы обработки данных называют сериализацией и десериализацией. Сериализация и десериализация JSON в сценарии веб-API обычно требует гораздо меньше работы, что, в свою очередь, приравнивается к лучшей производительности и меньшему количеству циклов вычисления. Это одна из причин, почему веб-API отлично подходят для передачи информации на мобильных устройствах и планшетах; в отличие от настольных компьютеров и служб, где они имеют ограниченные среды обработки. Веб-службы облегчают взаимодействие между двумя системами и почти всегда зависят от интерфейса, подобного XML-RPC. SOAP - преемник XML-RPC, определяет упомянутый выше обмен на основе XML и более связан с архитектурой клиент - сервер [17].

На данный момент веб-сервисы являются сервисом интеграции от одного устройства к другому; они обмениваются данными через Интернет и оптимизированы для связи между машинами, что означает, что машиночитаемые файлы и форматы (например, XML) легко переносятся. API-интерфейсы представляют собой программные интерфейсы с абстрактным набором функций для доступа к веб-приложениям.

Рассмотрев различные факторы, влияющие на выбор средства интеграции и изучив возможные методы интеграции специализированных информационных систем, можно сделать вывод, что для достижения поставленной цели в работе необходимо рассматривать интеграцию с помощью web-API. Выбор данного метода обеспечивает доступную и быструю разработку системы интеграции, а также обслуживание с использованием минимальных, как трудовых, так и финансовых затрат, за счет использования в разработке простого интерфейса управления информацией без использования дополнительных внутренних прослоек.

## 2 Проектирование интеграции информационных систем

### 2.1 Информационная модель интеграции систем

Перед тем, как приступить к проектированию интеграции специализированных информационных систем ИСОУ «Виртуальная школа» и «Парус 8» необходимо построить информационную модель «Как должно быть» (рисунок 1):

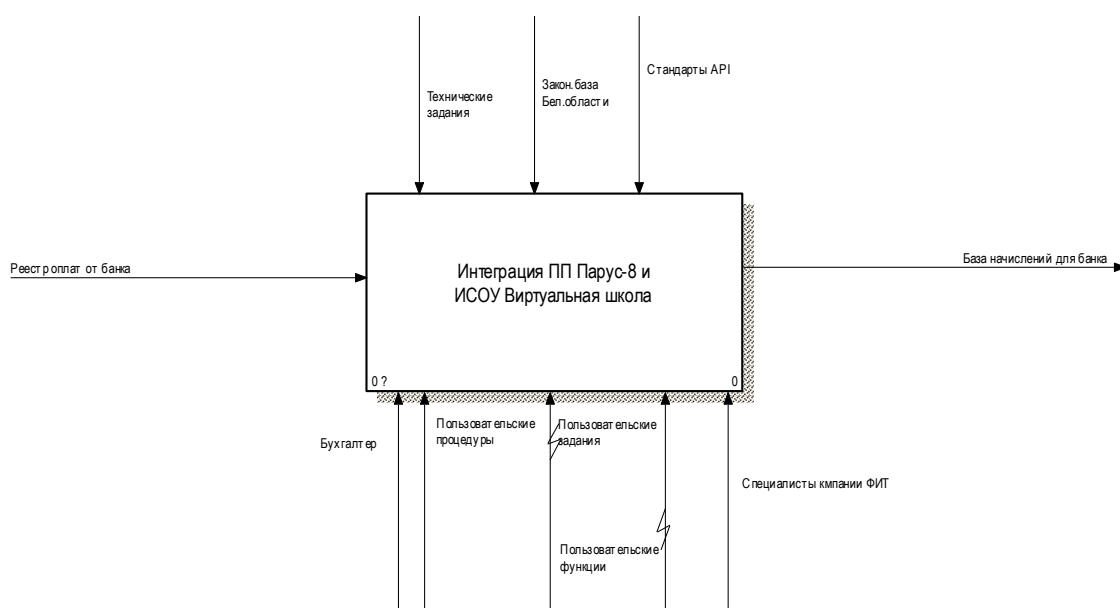


Рисунок 1 – Информационная модель «Как должно быть»

Процесс «Интеграция ПП Парус-8 и ИСОУ Виртуальная школа» имеет входные данные в виде реестров оплат от банка.

К выходным данным процесса относится база начислений для банка.

Управление включает в себя следующее: законодательная база Белгородской области; технические задания, стандарты API.

К механизму, в процессе «Интеграция ПП Парус-8 и ИСОУ Виртуальная школа» относится: бухгалтер, пользовательские задания, пользовательские функции, пользовательские процедуры и специалисты компании «ФИТ».

## Декомпозиция процесса «Интеграция ПП Парус-8 и ИСОУ Виртуальная школа» выглядит следующим образом (рисунок 2):

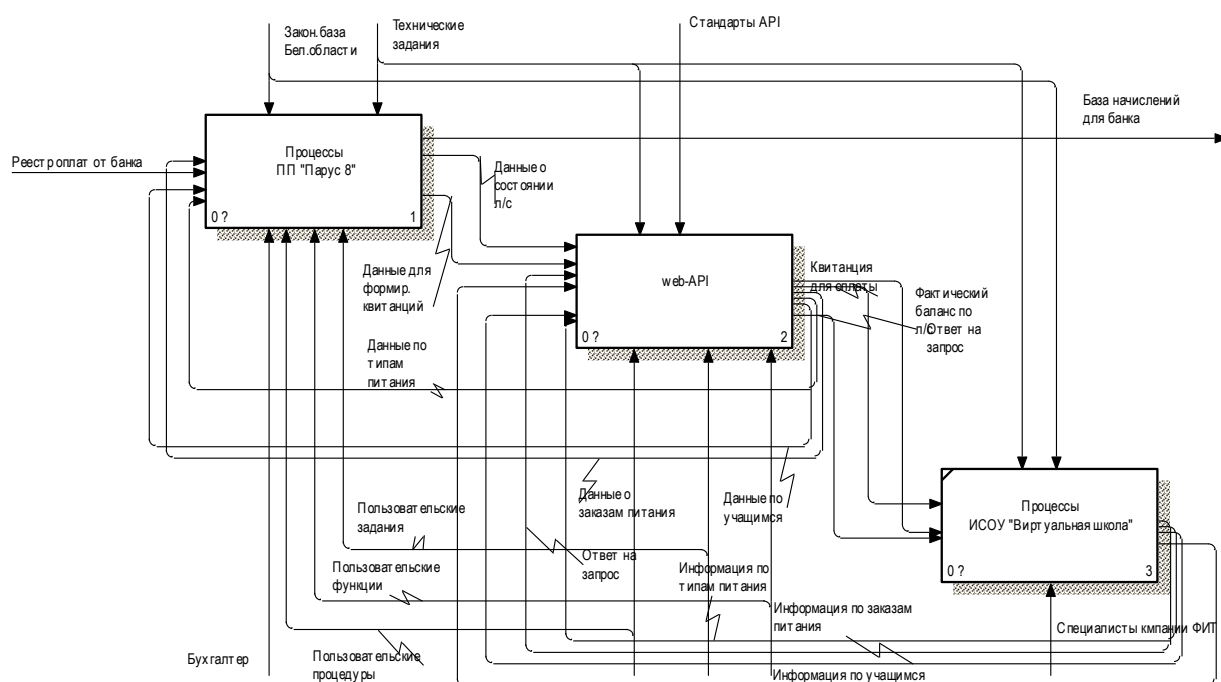


Рисунок 2 – Декомпозиция процесса «Интеграция Парус-8 и ИСОУ Виртуальная школа»

Декомпозиция процесса «Интеграция Парус-8 и ИСОУ Виртуальная школа» включает в себя следующие процессы: «ПП «Парус 8», «web-API», «ИСОУ «Виртуальная школа».

К входным данным процесса «ПП «Парус 8» относится реестры оплат от банка, данные по типам питания, данные по учащимся и данные по заказам питания.

К выходным данным процесса «ПП «Парус 8» относится база начислений для банка, данные о состоянии лицевого счета, данные для формирования квитанций.

Управление включает в себя законодательную базу Белгородской области, и технические задания.

Механизмом, в процессе «ПП «Парус 8» выступает бухгалтер, пользовательские задания, пользовательские функции и пользовательские процедуры.

Процесс «web-API» имеет входные данные в виде данных о состоянии лицевого счета, данных для формирования квитанций, информации по типам питания, информации об учащихся и информации о заказах питания, и ответа на запрос от ИСОУ «Виртуальная школа».

К выходным данным в процессе «web-API» относятся квитанции на оплату, фактический баланс по лицевым счетам, ответ на запрос, данные по учащимся, данные по типам питания и данные о заказах питания.

Управлением в процессе «web-API» выступают технические задания и стандарты API.

К механизму в процессе «web-API» относятся пользовательские задания, пользовательские функции и пользовательские процедуры.

Процесс «ИСОУ «Виртуальная школа» имеет следующие входные данные: квитанции на оплату, фактический баланс по лицевым счетам, ответ на запрос.

К выходным данным относится информация по учащимся, информация по типам питания, информация по заказам питания и ответ на запрос от web-API.

Управлением в процессе «ИСОУ «Виртуальная школа» выступает законодательная база Белгородской области и технические задания.

Механизмом в данном процессе являются специалисты компании «ФИТ».

Декомпозиция процесса «Парус 8» представлена на рисунке 3.

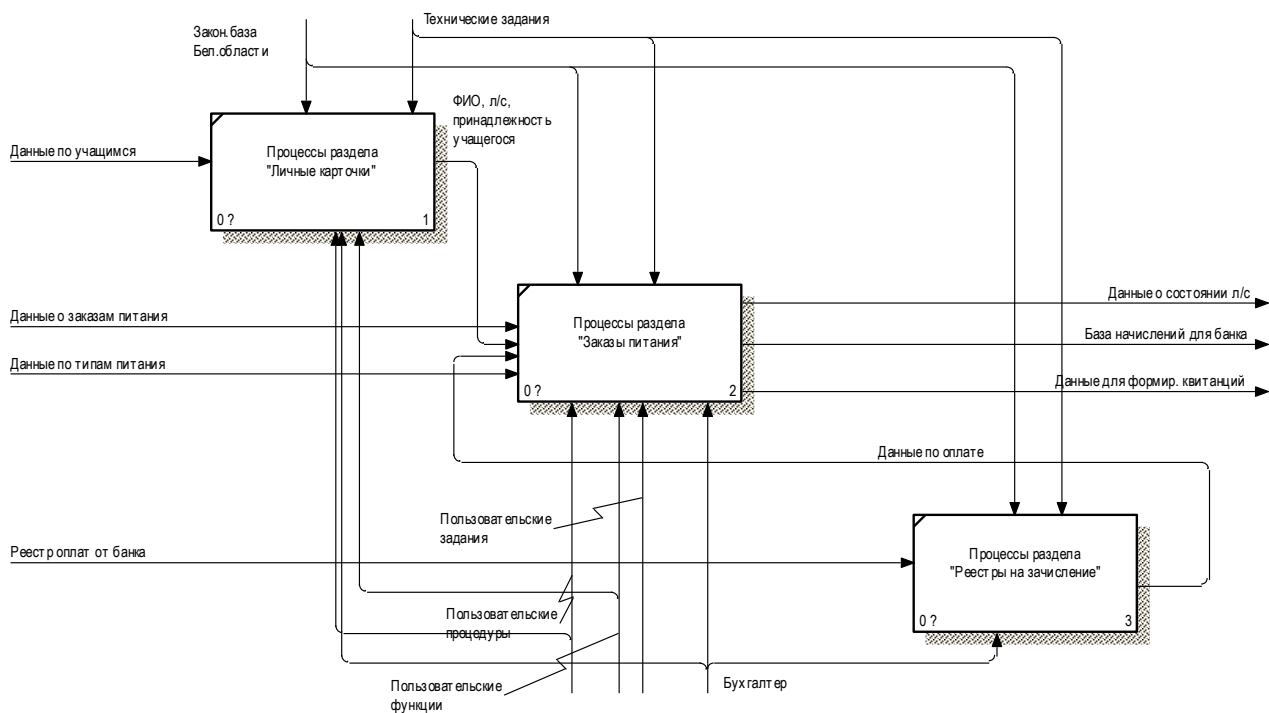


Рисунок 3 – Декомпозиция процесса «ПП "Парус 8»

Декомпозиция представлена в виде трех процессов: «Раздел «Личные карточки», «Раздел «Заказы питания» и «Раздел «Реестры на зачисление».

Процесс «Раздел «Личные карточки» имеет выходные данные в виде данных по учащимся. К выходным данным относятся ФИО, лицевой счет и принадлежность учащегося.

Управлением в процессе выступает законодательная база Белгородской области и технические задания. К механизму относятся бухгалтер, пользовательские задания, пользовательские функции и пользовательские процедуры.

В процессе «Раздел «Заказы питания» к входным данным относятся данные о заказах питания, данные по типам питания, ФИО, лицевой счет и принадлежность учащегося, и данные по оплате. Выходные данные в этом процессе заключаются в данных о состоянии лицевого счета, базе начислений для банка и данных для формирования квитанций.

К управлению в процессе «Раздел «Заказы питания» относится законодательная база Белгородской области и технические задания. К

механизму - бухгалтер, пользовательские задания, пользовательские функции и пользовательские процедуры.

Процесс «Раздел «Реестры на зачисление» имеет входные данные в виде реестра оплат от банка. К выходным данным относятся данные по оплате за питание учащихся.

Управлением в процессе «Реестры на зачисление» выступает законодательная база Белгородской области и технические задания. Механизмом выступает бухгалтер.

Декомпозиция процесса «web-API» изображена на рисунке 4.

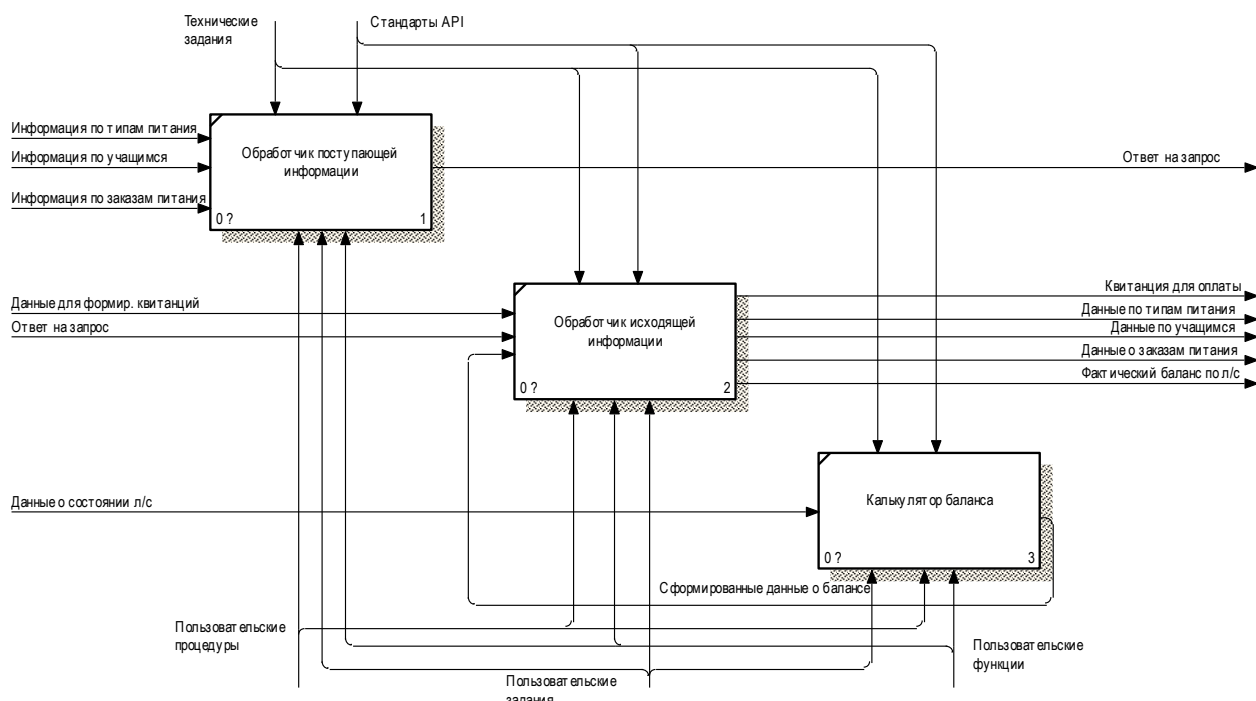


Рисунок 4 – Декомпозиция процесса «Web-API»

Декомпозиция процесса «Web-API» включает следующие процессы: «Обработчик поступившей информации», «Обработчик исходящей информации» и «Калькулятор баланса».

Процесс «Обработчик поступившей информации» имеет входные параметры такие как информация по типам питания, информация по учащимся, информация по заказам питания. К выходным параметрам этого процесса относится ответ на запрос.

К управлению в процессе «Обработчик поступившей информации» относятся технические задания и стандарты API. Механизмом в процессе выступают пользовательские задания, пользовательские функции и пользовательские процедуры.

В процессе «Обработчик исходящей информации» входными данными являются данные для формирования квитанций и ответ на запрос. К выходным данным относятся квитанции для оплаты, данные по типам питания, данные по заказам питания, данные по учащимся и фактический баланс по лицевому счету.

Управлением и механизмом выступает тоже, что и в процессе «Обработчик поступившей информации».

Процесс «Калькулятор баланса» имеет входной параметр в виде данных о состоянии лицевого счета. К выходным параметрам относятся сформированные данные о балансе. К управлению в процессе «Калькулятор баланса» относятся технические задания и стандарты API. Механизмом в процессе выступают пользовательские задания, пользовательские функции и пользовательские процедуры. Таким образом, была изложена модель «Как должно быть» процесса «Интеграция Парус-8 и ИСОУ Виртуальная школа».

## 2.2 Проектирование разделов интеграции систем

Согласно, техническому заданию заказчика, см. приложение А, необходимо разработать раздел «Заказы питания», раздел «Журнал взаимодействия» и интеграцию специализированных информационных системам.

Раздел «Заказы питания», который должен отражать информацию из штатного раздела «Личные карточки» в частности ФИО, лицевой счет, дату начала и дату конца действия, подведомственное учреждение и класс, в котором числится ученик. Также, раздел «Заказы питания» должен хранить

информацию по типам питания, начислению и списанию денежных средств и оплатах.

Раздел «Журнал взаимодействия» для хранения и обработки данных в результате интеграции с «ИСОУ «Виртуальная школа» с помощью API сервиса.

Интеграцию специализированных информационных системам «Парус 8» и «ИСОУ «Виртуальная школа» с помощью веб-сервиса по средствам API.

### 2.2.1 Проектирование раздела «Заказы питания»

Описание раздела.

Раздел «Заказы питания» должен вызываться в меню «Учет» и иметь каталожную структуру, делиться по учреждениям.

Заголовок раздела должен иметь следующие поля:

- «Личная карточка» - ссылка на раздел «Личные карточки»;
- «ФИО» - заполняется автоматически из полей Фамилия, Имя, Отчество раздела «Личные карточки» по связи с номером личной карточки;
- «Номер лицевого счета» - заполняется автоматически из поля «Лицевой счет» раздела «Личные карточки» по связи с номером личной карточки;
- «Учреждение» - ссылка на заголовок раздела «Учреждения»;
- «Класс» - ссылка на спецификацию «Группы» раздела «Учреждения»;
- «Региональный бюджет» - числовое поле, рассчитывается путем сложения значений поля «Региональный бюджет» спецификации «Питание»;
- «Муниципальный бюджет» - числовое поле, рассчитывается путем сложения значений поля «Региональный бюджет» спецификации «Питание»;
- «Родительская оплата» - числовое поле, рассчитывается путем сложения значений поля «Родительская оплата» спецификации «Питание».



Спецификация раздела «Заказы питания» должна иметь четыре вкладки: «Питание», «Начисления», «Оплаты» и «Расчет за месяц».

Спецификация «Питание» должна состоять из следующих полей:

- «Дата» - поле формата ДД.ММ.ГГГГ;
- «Наименование блюда» - значение соответствует полю «Примечание» из доп.словаря «Виды блюд»;
- «Тип питания» - значение из доп.свойства «Тип питания» соответствующего «Вида блюда»;
- «Региональный бюджет» - значение из доп.свойства «Региональный бюджет» соответствующего «Вида блюда»;
- «Муниципальный бюджет» - значение из доп.свойства «Муниципальный бюджет» соответствующего «Вида блюда»;
- «Родительская оплата» - значение из доп.свойства «Родительская оплата» соответствующего «Вида блюда».

В спецификации «Питание» должны быть стандартные действия: «Добавить», «Изменить», «Удалить».

Спецификация «Начисления» формируется на основании спецификации «Питание», должна состоять из следующих полей:

- «Наименование» - поле соответствует полю «Наименование» спецификации «Питание»;
- «Месяц» - строковое поле, заполняется из параметра Расчетный период;
- «Год» - числовое поле, заполняется из параметра Расчетный период;
- «Количество» - числовое поле, значение поля соответствует количеству записей данного наименования в спецификации «Питание» в текущем расчетном периоде;
- «Региональный бюджет» - числовое поле, поле вычисляется путем суммирования значений поля «Региональный бюджет» в

спецификации «Питание» да данному наименованию в данном расчетном периоде;

– «Муниципальный бюджет» - числовое поле, поле вычисляется путем суммирования значений поля «Муниципальный бюджет» в спецификации «Питание» да данному наименованию в данном расчетном периоде;

– «Родительская оплата» - числовое поле, поле вычисляется путем суммирования значений поля «Родительская оплата» в спецификации «Питание» да данному наименованию в данном расчетном периоде.

В спецификации «Начисления» стандартные действия отсутствуют.

Спецификация «Оплаты» должна имеет следующие поля: «Год», «Месяц», «Сумма», «Дата платежа», «Тип оплаты», «Погашение задолженности», «Документ-основание», «Тип документа-основания», «Номер документа», «Дата документа-основания», «Примечание».

В спецификации «Оплаты» должны быть стандартные действия: «Добавить», «Изменить», «Удалить».

Сумма в поле «Входящий остаток» - поле заполняется автоматически, путем переноса значения из поля «Исходящий остаток» прошлого периода.

Сумма в поле «Начислено» - поле заполняется автоматически из поля «Начислено» (Родительская оплата) спецификации «Начисления»;

Сумма в поле «Оплачено» - поле заполняется автоматически, путем сложения значений поля «Сумма» в спецификации «Оплаты», у которых Тип оплаты - Оплата.

Сумма в поле «Возвращено» - поле заполняется автоматически, путем сложения значений поля «Сумма» в спецификации «Оплаты», у которых Тип оплаты - Возврат.

Сумма в поле «Исходящий остаток» - значение поля рассчитывается автоматически путем вычитания значения из поля «Оплачено» значение поля «Начислено».

В спецификации «Расчет за месяц» стандартные действия отсутствуют.

Действия в разделе.

Заголовок документа должен содержать стандартные действия «Добавить», «Размножить», «Исправить», «Удалить».

Данные в разделе должны отображаться в рамках одного периода (месяца). Создать действие «Сменить расчетный период».

При выполнении действия отображается форма для задания нового расчетного периода (месяц, год). После выбора периода и нажатия ОК изменяются данные, отображаемые в спецификациях "Начисления" и "Оплаты".

В разделе «Заказы питания» отображаются только те записи, в личных карточках которых срок действия включает в себя хотя бы один день текущего расчетного периода.

### 2.2.2 Проектирование раздела «Журнал взаимодействия»

Раздел должен вызываться в меню «Учет», иметь каталожную структуру и делиться по учреждениям.

Заголовок раздела должен содержать следующие поля:

- «Пользователь» - учетное имя пользователя, по которым выполняется пользовательское задание;
- «Вид операции» - текстовое поле, может принимать следующие значения: «Получение данных о заказах питания», «Получение данных об учениках», «Передача данных по оплатам», «Передача квитанций», «Получение данных о типах питания»;
- «Идентификатор документа» - числовое поле, значение номера обработки данных на сервисе;
- «Дата начала действия» - поле в формате ДД.ММ.ГГГГ ЧЧ.ММ.СС, дата и время запуска пользовательского задания;

- «Дата окончания действия» - поле в формате ДД.ММ.ГГГГ ЧЧ.ММ.СС, дата и время окончания работы пользовательского задания;
- «Количество сообщений» - числовое поле, для записи количества сообщений, возникших при передаче данных с сервиса;
- «Количество ошибок» - числовое поле, для записи количества ошибок, возникших при передаче данных с сервиса;
- «Время выполнения» - поле в формате ЧЧ.ММ.СС, для записи фактического времени выполнения пользовательского задания;
- «Дата документа» - поле в формате ДД.ММ.ГГГГ ЧЧ.ММ.СС, для записи времени, когда поступили данные от ИСОУ «Виртуальная школа».

Спецификация раздела «Журнал взаимодействия» должна иметь следующие поля:

- «Тип сообщения» - текстовое поле, которое может принимать следующие значения: «Сообщение», «Ошибка»;
  - «Вид операции» - текстовое поле, значение соответствует полю «Вид операции» заголовка раздела;
  - «Учреждение» - текстовое поле, для записи Id учреждения, по которому возникла ошибка на этапе обработки данных;
  - «Лицевой счет» - текстовое поле, для записи лицевого счета учащегося, по которому возникла ошибка на этапе обработки данных;
  - «Текст ошибки» - текстовое поле, для записи полного текста ошибки при обработке данных.
- Раздел «Журнал взаимодействия» должен иметь следующие действия:
- «Получить тип питания» - действие по которому выполняется обращение на сервис, для получения данных по типам питания;
  - «Отправить квитанции» - действие по которому формируется массив данных по квитанциям и отправляется на сервис;

- «Отправка оплат» - действие по которому формируется массив данных по оплатам и отправляется на сервис;
- «Синхронизировать» - действие, по которому идет обращение к сервису и выполняются пользовательские задания по загрузке данных;
- «Отработать» - действие повторной отработки записей, по котором были ошибки в предыдущих загрузках.

## 2.3 Проектирование интеграции информационных системам «Парус 8» и «ИСОУ «Виртуальная школа» с помощью веб-сервиса по средствам API

### 2.3.1 Загрузка данных в раздел «Заказы питания»

Параметры загрузки.

Данные по заказам питания передаются в следующем формате, таблица 1.

Таблица 1 - Формат данных заказов питания

Параметр	Тип данных	Размерность	Наименование
schoolId	Int		Идентификатор школы
classId	Int		Идентификатор класса
checkingAccount	String		Лицевой счет
dateTime	String	[dd.MM.yyyy HH:mm:ss]	Дата и время заказа
studentFullName	String		ФИО ученика
foodTypeId	Int		Идентификатор типа питания
regionalBudget	Int		Региональный бюджет
municipalBudget	Int		Муниципальный бюджет
parentalPayment	Int		Родительская оплата

Данные загружаются из сервиса по API [21]. Параметры API: PUT/provders/{providerId}/food\_orders

Тело запроса объектов в json описано в приложении Б.

Данные должны загружаться в раздел «Заказы питания».

Алгоритм добавления данных:

На первом шаге (а) нужно в разделе «Заказы питания» осуществлять поиск по полю «Номер лицевого счета», значение поля «Номер лицевого счета» равно значению поля «checkingAccount». Иначе, в п.д.

На втором шаге (b) смотрим поле «dateTime». Если значение поля меньше «текущая дата минус один», то по записям найденных в п.1 обращаемся к спецификации «Заказы питания (Питание)» и удаляем записи, у которых значение поля «Дата» в спецификации равно значению поля «dateTime». Иначе, у найденной записи в п. а смотрим поле «Признак активности». Если «Признак активности» = Да, то переходим к спецификации «Заказы питания (Питание)» пункт с. Иначе переходим к п.д.

На третьем шаге (с) в спецификацию «Заказы питания (Питание)» добавляем записи, где

- в поле «Дата» записываем значение из поля «dateTime»;
- в поле «Наименование блюда» записываем значение, у которого значение поля «foodTypeId» равно значению «Мнемокода» словаря «Заказы питания (Блюда)».

- поля «Тип питания», «Региональный бюджет», «Муниципальный бюджет» и «Родительская оплата» заполняются соответственно из словаря «Заказы питания (Блюда)» спецификации «Периоды действия» в зависимости от заполнения поля «Наименование блюда».

На четвертом шаге (d) в заголовке раздела «Заказы питания» должна добавляться новая запись, в которой в поле «Период» записываем период, который определяется по полю «dateTime», например, «dateTime» = 05.09.2017, то период – Сентябрь, записываем в формате 09/2017.

Личная карточка должна выбираться по лицевому счету. Значение поля «checkingAccount» равно значению поля «Номер лицевого счета» в разделе «Личные карточки».

Значение поля «Учреждение», «Класс» и «Признак активности» заполняются автоматически из спецификации «История» раздела «Личные карточки» по полю «Признак активности» = «Да».

Поля «Тип питания», «Региональный бюджет», «Муниципальный бюджет» и «Родительская оплата» заполняются соответственно из словаря «Заказы питания (Блюда)» спецификации «Периоды действия» в зависимости от заполнения поля «Наименование блюда».

На пятом шаге (е) должна заполняться спецификация «Заказы питания (Питание)». Алгоритм заполнения спецификации описан в пункте с.

### 2.3.2 Загрузка данных по учащимся

Параметры загрузки.

Данные по учащимся передаются в следующем формате, таблица 2.

Таблица 2 - Формат данных по учащимся

Параметр	Тип данных	Размерность	Наименование
schoolId	Int		ID школы
schoolName	String		Наименование школы
classId	Int		ID класса
className	String		Наименование класса
checkingAccount	String		Лицевой счет
studentFullName	String		ФИО ученика
recommendedPayment	Int		Рекомендованный платеж
studentActive	Boolean		Признак активности ученика
enrollmentDate	Date	[ddMMyyyy]	Дата зачисления
dismissalDate	Date	[ddMMyyyy]	Дата выбытия

Данные загружаются из сервиса по API [22]. Параметры API: PUT/provders/{providerId}/students

Тело запроса объектов в json описано в приложении В.

Данные должны загружаться в раздел «Контрагенты» и «Личные карточки». Алгоритм добавления данных:

В разделе «Контрагенты» осуществляем поиск в поле «Мнемокод» по значению «checkingAccount»

если запись контрагента найдена, то обновляем поля: Фамилия («studentFullName»), Имя («studentFullName»), Отчество («studentFullName»), (Наименование studentFullName).

Далее определяем наличие личной карточки по checkingAccount

если запись существует, то в спецификацию «История» добавляем запись, у которой:

- «Дата обновления» - дата загрузки данных;
- «Дата постановки на учет» - значение поля «enrollmentDate»;
- «Дата снятия с учета» - значение поля «dismissalDate»
- «Наименование школы» - определяем по полю «schoolId».

Значение поля «schoolId» ищем в разделе «Учреждения» в поле «Мнемокод».

Если запись найдена, то в поле «Наименование школы» записываем мнемокод контрагента найденной записи. Иначе, поле остается пустым.

– «Наименование класса» - определяем по полю «classId». Значение поля «classId» ищем в спецификации «Группы» учреждения, выбранного в поле «Наименование школы». Если соответствия не найдено, то поле остается пустым.

– «Признак активности» - значение соответствует полю «studentActive». Если значение поля «studentActive» = true, то «Признак активности» = Да, иначе «Признак активности» = Нет.

Если личная карточка не определена, то добавляем (описание см.ниже).

После этого переходим к следующей записи в загружаемом файле.

если запись контрагента не найдена, то должна создаваться новая запись, в которой:

- поле «Тип» - «Физическое лицо»;



- поле «Фамилия» - значение 1 подстроки поля «studentFullName» загружаемого файла, (разделитель пробел);
- поле «Имя» - значение 2 подстроки поля «studentFullName» загружаемого файла, (разделитель пробел);
- поле «Отчество» - значение 3 подстроки поля «studentFullName» загружаемого файла, (разделитель пробел);
- поле «Мнемокод» - значение поля «checkingAccount»;
- поле «Наименование» - значение поля «studentFullName»
- добавление личной карточки
- после добавления контрагента переходим к добавлению личной карточки (в разделе «Личные карточки»):
- поле «Номер» - текущий номер по порядку;
- поле «Контрагент» - записываем контрагента, добавленного в раздел «Контрагенты» (по значению «checkingAccount»);
- поле «Номер лицевого счета» – в префикс записываем «ВШ», в номер записываем значение поля «checkingAccount» загружаемого файла;
- поле «Дата постановки на учет» - значение поля «enrollmentDate»;

Далее заполняем спецификацию «История»:

добавляем запись, у которой:

- «Дата обновления» - дата загрузки данных;
- «Дата постановки на учет» - значение поля «enrollmentDate»;
- «Дата снятия с учета» - значение поля «dismissalDate»
- «Наименование школы» - определяем по полю «schoolId».

Значение поля «schoolId» ищем в разделе «Учреждения» в поле «Мнемокод». Если запись, найдено, то в поле «Наименование школы» записываем мнемокод контрагента найденной записи. Иначе, поле остается пустым.

- «Наименование класса» - определяем по полю «classId». Значение поля «classId» ищем в спецификации «Группы» учреждения, выбранного в

поле «Наименование школы». Если соответствия не найдено, то поле остается пустым.

– «Признак активности» - значение соответствует полю «studentActive». Если значение поля «studentActive» = true, то «Признак активности» = Да, иначе «Признак активности» = Нет.

### 2.3.3 Выгрузка квитанций по лицевым счетам

Параметры выгрузки.

Данные выгружаются с сервиса по API. Параметры API: PUT/food\_providers/{providerId}/receipts

Тело запроса объектов в json описано в приложении Г.

Данные для выгрузки:

В поле «Получатель платежа» записываем значение из поля "Учреждение" раздела «Заказы питания»

В поле «ЛС» записываем значение, которое определяется по связи: «Учреждения»- «Контрагенты» - «Реквизиты лицевых счетов в финансовых учреждениях». (Определяем поле «Номер счета». В спецификации «Реквизиты лицевых счетов в финансовых учреждениях» ищем строку у которой поле «Код строки»= «\*ицевой\*», в найденной строке считываем значение «Номер счета» и записываем его в поле «ЛС» ).

В поле «ИНН» записываем значение из поля «Идентификационный номер налогоплательщика» заголовка раздела «Контрагенты» по значению указанного в поле «Получатель платежа».

В поле «ОКТМО» записываем значение из поля «Код ОКТМО» заголовка раздела «Контрагенты» по значению указанного в поле «Получатель платежа».

Описание заполнения полей "Счет", "Банк", "БИК".

Значение полей "ЛС", "Счет", "Банк" и "БИК" брать из строки, указанной в поле "Банковские реквизиты контрагента" в разделе "Учреждения".

Если в поле "Банковские реквизиты контрагента" указано значение, у которого заполнено поле "Казначейство", то по связи: «Казначейство» - «Реквизиты казначейства в банке». В поле «Счет» записываем значение из поля «Счет», в поле «Банк» записываем значение из поля «Наименование», в поле «БИК» записываем значение из поля «Код банка (БИК)».

Если в поле "Банковские реквизиты контрагента" указано значение, у которого заполнено поле "Казначейство", то в поле «ЛС» записываем значение из поля «Номер счета» по связи: «Контрагент» - «Реквизиты лицевых счетов в банковских учреждениях» - поле «Номер счета» (вкладка Реквизиты).

Иначе, л/с организации выбираем из поля "Лицевой счет" заголовка раздела "Учреждения". Значение полей "Счет", "Банк" и "БИК" брать из строки, указанной в поле "Банковские реквизиты контрагента" в разделе "Учреждения".

В Л/с организации записывать значение поля "Номер счета" строки реквизита, которая указана в Учреждении только тогда, когда у строки реквизита указанной в Учреждении заполнена вкладка "Казначейства".

Иначе, л/с организации выбираем из поля "Лицевой счет" заголовка раздела "Учреждения".

Иначе, в спецификации «Реквизиты лицевых счетов в финансовых учреждениях» ищем строку, у которой поле «Код строки» = «азначейс», в найденной строке считываем значение «Номер счета» и записываем его в поле «Счет»; в поле «Банк» записываем мнемокод учреждения, записанного в поле «Банковское учреждение»; в поле «БИК» записываем значение из поля «БИК» вкладка «Банк».

Если найдено больше одного значения, то выбираем первое найденное. Если данных не найдено, поля в квитанции оставляем пустые.

В поле «КБК» записываем значение, указанное в параметре экранной формы «Код бюджетной классификации».

В поле «Код причины постановки на учет» записываем значение из поля «Код причины постановки на учет» заголовка раздела «Контрагенты» по контрагенту, указанному в поле «Получатель платежа».

В поле «Назначение платежа» записываем выражение «Код субсидии» плюс значение из параметра экранной формы «Код субсидии».

В поле «Лицевой счет учащегося» записываем значение из заголовка раздела «Заказы питания» из поля «Номер лицевого счета».

В поле «ФИО» записываем значение из заголовка раздела «Заказы питания» из поля «ФИО».

В поле «Класс» записываем значение из заголовка раздела «Заказы питания» из поля «Класс».

В поле «Месяц» записываем значение из заголовка раздела «Заказы питания» из поля «Период».

Пример квитанции представлен на рисунке 5.


	Получатель платежа: Шк Ракитянская №3, ЛС: 20266871041, ИНН: 3116003574, ОКТМО: 14648151, КПП:		
	Банк: УФК, БИК: 041403001, Счет: 40701810314031000082 КБК: 871000000000000000130		
КВИТАНЦИЯ	Адрес:	Лицевой счет ребенка	<b>00973286</b>
	Назначение платежа: Код субсидии 3401602 л/с 20266871041		
	Ф.И.О.: Иванова Олеся Валерьевна		
	вид платежа	Класс	Месяц
	Оплата за питание	10	12/2017
			Сумма
			0,00

Рисунок 5 – Пример квитанции для выгрузки

Формат данных для ДШК представлен в таблице 3.

Таблица 3 - Формат данных квитанции

<b>Псевдоним</b>	<b>К-во символов</b>	<b>Наименование поля в файле выгрузки</b>
Name	160	Получатель платежа+(ЛС)
PersonalAcc	20	Счет
BankName	100	Банк
BIC	9	БИК
CorrespAcc	20	В файле выгрузки отсутствует. Заполняется нулями. Обязателен для Сбербанка
PayeeINN	11	ИНН
PersAcc	10	Лицевой счет ребенка
TechCode	60	Значение равно 02

#### 2.3.4 Выгрузка данных по состоянию баланса лицевого счета

Выгрузка осуществляется по всем каталогам и по всем записям в разделе «Заказы питания».

Формат данных для выгрузки оплат представлен в таблице 4.

Таблица 4 - Формат данных по балансу

<b>Параметр</b>	<b>Тип данных</b>	<b>Размерность</b>	<b>Наименование</b>
checkingAccount	String		лицевой счет
dateTime	String	[dd.MM.yyyy HH:mm:ss]	дата и время совершения операции
balanceBefore	Int		баланс до начала операции
incoming	Int		поступило денег
writeOff: Int	Int		списано денег
balance	Int		баланс после операции

Данные выгружаются с сервиса по API. Параметры API: PUT/food\_providers/{providerId}/ payments

Тело запроса объектов в json описано в приложении Д.

Заполнение полей для выгрузки данных:

- `checkingAccount` - «Лицевой счет» - значение из поля «Номер лицевого счета» раздела «Заказы питания»;
- `dateTime` - дата выгрузки, формат ДД.ММ.ГГГГ;
- `incoming` - «Оплачено» - значение из поля «Оплачено» вкладки «Расчет за месяц» раздела «Заказы питания»;
- `writeOff` - «Начислено» - значение из поля «Начислено» вкладки «Расчет за месяц» раздела «Заказы питания»;
- `balance` - «Исходящий остаток» - значение из поля «Исходящий остаток» вкладки «Расчет за месяц» раздела «Заказы питания».

В данном разделе была смоделирована интеграция информационных систем и построена диаграмма «Как должно быть». Данная диаграмма показывает взаимодействие всех объектов ПП «Парус 8» и ИСОУ «Виртуальная школа» при интеграции. На основе данной модели были спроектированы объекты в ПП «Парус 8» и web-сервис для обмена данными с ИСОУ «Виртуальная школа».

Далее следует описать процесс программной реализации интеграции специализированных информационных систем.

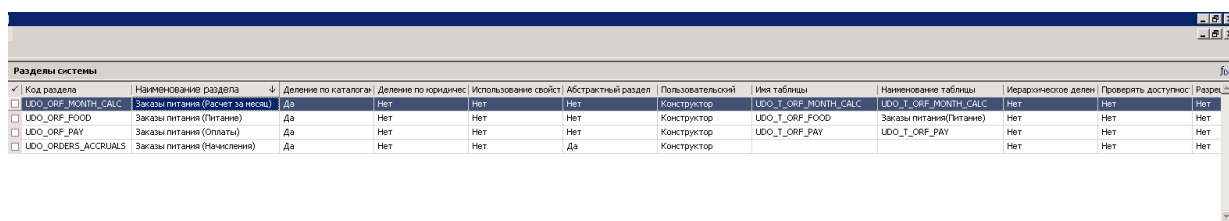
### 3 Программная реализация интеграции информационных систем

#### 3.1 Разработка в ПП «Парус 8»

Разработка разделов в ПП «Парус 8» производится в интегрированной среде разработки - Oracle SQL Developer.

Oracle PL/SQL Developer - это бесплатная интегрированная среда разработки, которая упрощает процесс разработки и управления базой данных Oracle как в традиционных, так и в облачных развертываниях. PL/SQL Developer - комплексное решение для моделирования данных и платформ интеграции Oracle со сторонними базами данных.

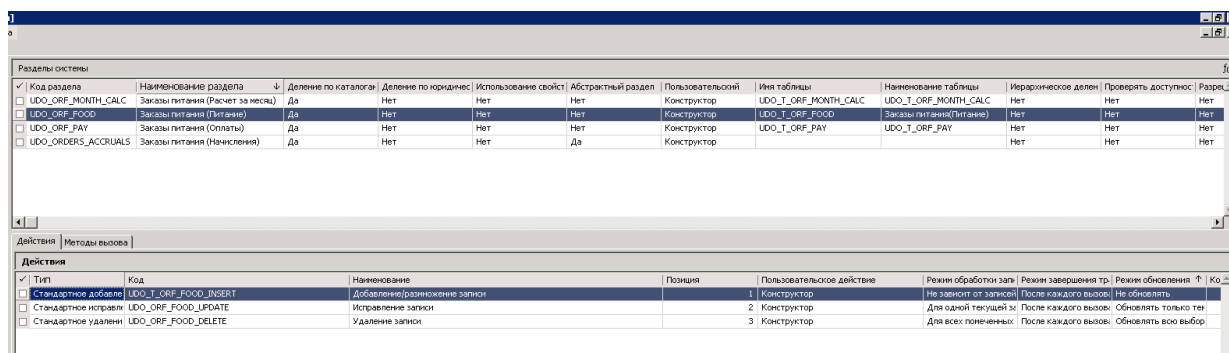
Разработка разделов производится в модуле «Администратор» ПП «Парус 8». Добавили структуру раздела «Заказы питания», рисунок 6.



Код раздела	Наименование раздела	Деление по каталогам	Деление по юридическим	Использование свойств	Абстрактный раздел	Пользовательский	Имя таблицы	Наименование таблицы	Иерархическое деление	Проверять доступность	Разрешить
<input checked="" type="checkbox"/>	UDO_ORF_MONTH_CALC	Заказы питания (Расчет за месяц)	Да	Нет	Нет	Конструктор	UDO_T_ORF_MONTH_CALC	UDO_T_ORF_MONTH_CALC	Нет	Нет	Нет
<input type="checkbox"/>	UDO_ORF_FOOD	Заказы питания (Питание)	Да	Нет	Нет	Конструктор	UDO_T_ORF_FOOD	Заказы питания(Питание)	Нет	Нет	Нет
<input type="checkbox"/>	UDO_ORF_PAY	Заказы питания (Оплаты)	Да	Нет	Нет	Конструктор	UDO_T_ORF_PAY	UDO_T_ORF_PAY	Нет	Нет	Нет
<input type="checkbox"/>	UDO_ORDERS_ACCRUALS	Заказы питания (Начисления)	Да	Нет	Нет	Конструктор			Нет	Нет	Нет

Рисунок 6 – Структура раздела «Заказы питания»

В спецификацию «Питание» и «Оплаты» добавили следующие действия, рисунок 7 - 8.



Тип	Код	Наименование	Позиция	Пользовательское действие	Режим обработки зап.	Режим завершения тр.	Режим обновления
<input checked="" type="checkbox"/>	UDO_ORF_INSERT	Добавление записей	1	Конструктор	Для одной текущей записи	После каждого вызова	Обновлять только те
<input type="checkbox"/>	UDO_ORF_UPDATE	Исправление записей	2	Конструктор	Для одной текущей записи	После каждого вызова	Обновлять только те
<input type="checkbox"/>	UDO_ORF_DELETE	Удаление записей	3	Конструктор	Для всех поименованных	После каждого вызова	Обновлять всю выборку

Рисунок 7 – Действия для спецификации «Питание»

Разделы системы												
<input checked="" type="checkbox"/>	Код раздела	Наименование раздела	Деление по каталогу	Деление по юридичес	Использование свойств	Абстрактный раздел	Пользовательский	Имя таблицы	Наименование таблицы	Иерархическое делен	Проводить доступнос	Разре
<input type="checkbox"/>	UDO_ORF_MONTH_CALC	Заказы питания (Расчет за месяц)	Да	Нет	Нет	Нет	Конструктор	UDO_T_ORF_MONTH_CALC	UDO_T_ORF_MONTH_CALC	Нет	Нет	Нет
<input type="checkbox"/>	UDO_ORF_FOOD	Заказы питания (Питание)	Да	Нет	Нет	Нет	Конструктор	UDO_T_ORF_FOOD	Заказы питания(Питание)	Нет	Нет	Нет
<input checked="" type="checkbox"/>	UDO_ORF_PAY	Заказы питания (Оплаты)	Да	Нет	Нет	Нет	Конструктор	UDO_T_ORF_PAY	UDO_T_ORF_PAY	Нет	Нет	Нет
<input type="checkbox"/>	UDO_ORDERS_ACCRUALS	Заказы питания (Начисления)	Да	Нет	Нет	Да	Конструктор			Нет	Нет	Нет

Действия								
<input checked="" type="checkbox"/>	Тип	Код	Наименование	Позиция	Пользовательское действие	Режим обработки зат	Режим завершения тр	Режим обновления
<input type="checkbox"/>	Стандартное добавлен	UDO_ORF_PAY_INSERT	Добавление/размещение оплаты		1 Конструктор	На момент от заказов	После каждого вызова	Не обновлять
<input type="checkbox"/>	Стандартное исправлени	UDO_ORF_PAY_UPDATE	Исправление записи		2 Конструктор	Для одной текущей за	После каждого вызова	Обновлять только те
<input type="checkbox"/>	Стандартное удалени	UDO_ORF_PAY_DELETE	Удаление записи		3 Конструктор	Для всех поменяных	После каждого вызова	Обновлять всю выбор

Рисунок 8 – Действия для спецификации «Оплаты»

Для раздела «Заказы питания» необходимо создать три таблицы данных: «UDO\_T\_ORF\_MONTH\_CALC», «UDO\_T\_ORF\_FOOD» и «UDO\_T\_ORF\_PAY».

Создание таблицы данных «UDO\_T\_ORF\_MONTH\_CALC» представлено на рисунках 9 – 12.

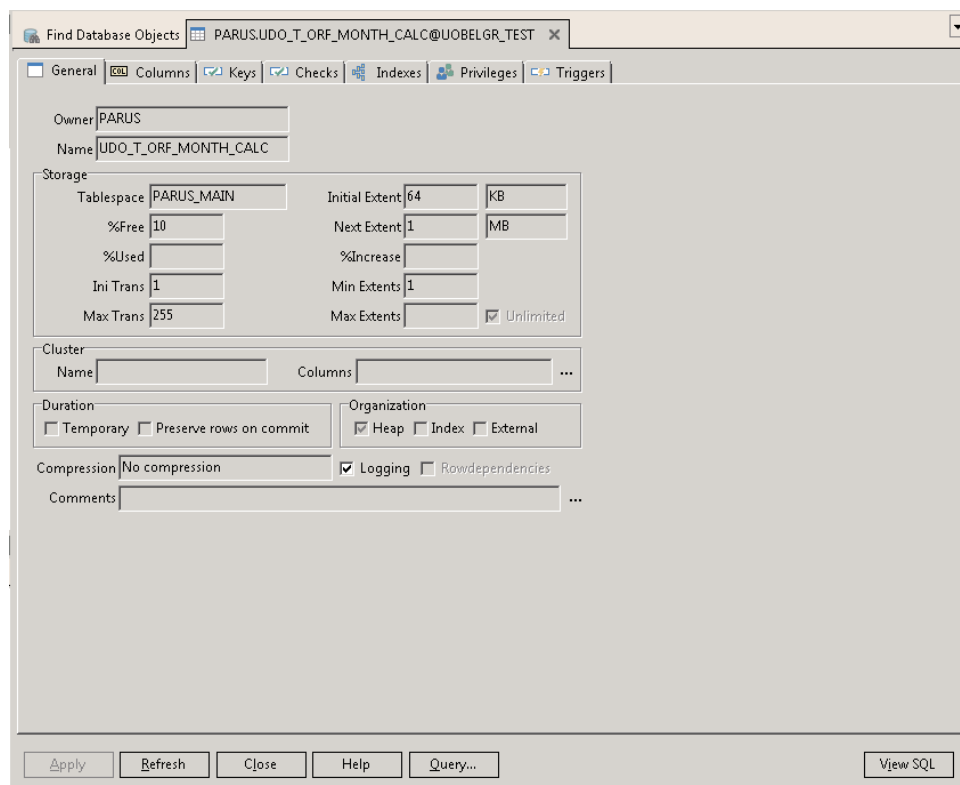


Рисунок 9 – Вкладка «Главная» таблицы «UDO\_T\_ORF\_MONTH\_CALC»



Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Storage	Comments
PRN	<input type="checkbox"/>	NUMBER(17)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
COMPANY	<input type="checkbox"/>	NUMBER(17)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
CODE	<input type="checkbox"/>	NUMBER(2)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
NAME	<input type="checkbox"/>	VARCHAR2(40)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
SUM	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	"PARUS"."UDO_F_ORF_MCALC'('PRN','COMPANY','CODE')		<input type="checkbox"/>		

Рисунок 10 – Вкладка «Колонки» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred
UDO_ORF_MONTH_CALC_COMPANY	Foreign	COMPANY	<input checked="" type="checkbox"/>	COMPANIES	RN	No action	<input type="checkbox"/>	<input type="checkbox"/>
UDO_ORF_MONTH_CALC_PRN	Foreign	PRN	<input checked="" type="checkbox"/>	UDO_T_ORDERS_FOOD	RN	Cascade	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 11 – Вкладка «Ключи» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

Owner	Name	Type	Columns	Compress	Logging	Prefix length	Invisible	Local	Reverse	Storage
PARUS	I_UDO_ORF_MONTH_CALC_PRN	Normal	PRN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri

Рисунок 12 – Вкладка «Индексы» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

Создание таблицы данных «UDO\_T\_ORF\_FOOD» представлено на рисунках 13 – 16.

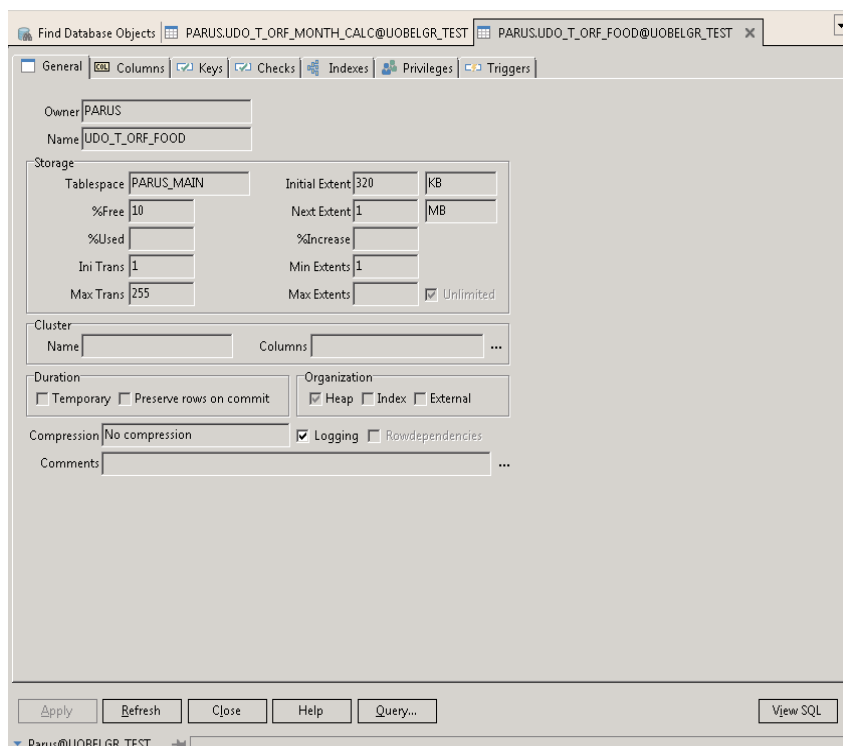


Рисунок 13 – Вкладка «Главная» таблицы «UDO\_T\_ORF\_FOOD»

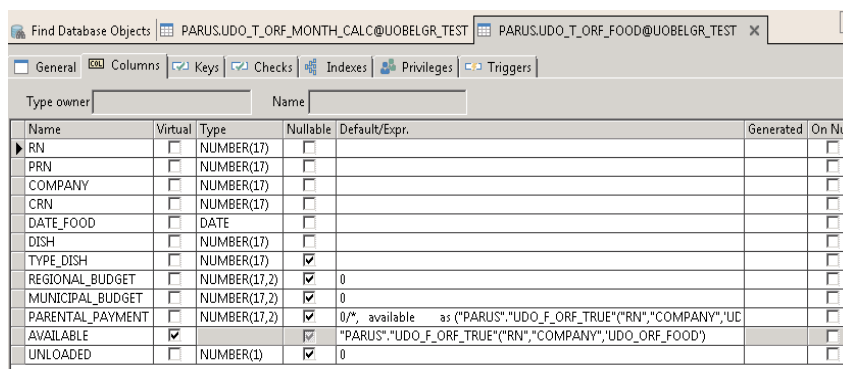


Рисунок 14 – Вкладка «Поля» таблицы «UDO\_T\_ORF\_FOOD»

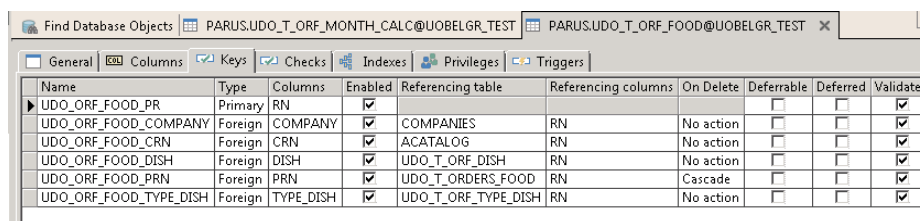


Рисунок 15 – Вкладка «Ключи» таблицы «UDO\_T\_ORF\_FOOD»

Owner	Name	Type	Columns	Compress	Logging	Prefix length	Invisible	Local	Reverse	Storage
PARUS	I_UDO_ORF_FOOD_COMPANY	Normal	COMPANY	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri
PARUS	I_UDO_ORF_FOOD_CRN	Normal	CRN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri
PARUS	I_UDO_ORF_FOOD_DISH	Normal	DISH	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri
PARUS	I_UDO_ORF_FOOD_PRN	Normal	PRN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri
PARUS	I_UDO_ORF_FOOD_TYPE_DISH	Normal	TYPE_DISH	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri
PARUS	UDO_ORF_FOOD_PR	Unique	RN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfri

Рисунок 16 – Вкладка «Индексы» таблицы «UDO\_T\_ORF\_FOOD»

Создание таблицы данных «UDO\_T\_ORF\_PAY» представлено на рисунках 17 - 21.

Owner: PARUS  
Name: UDO\_T\_ORF\_PAY

Storage:  
 Tablespace: PARUS\_MAIN  
 Initial Extent: 64 KB  
 %Free: 10  
 Next Extent: 1 MB  
 %Used:   
 %Increase:   
 Ini Trans: 1  
 Min Extents: 1  
 Max Trans: 255  
 Max Extents:  Unlimited

Cluster:  
 Name:   
 Columns: ...

Duration:  
 Temporary  Preserve rows on commit

Organization:  
 Heap  Index  External

Compression: No compression  Logging  Rowdependencies

Comments: ...

Buttons: Apply, Refresh, Close, Help, Query..., View SQL

Рисунок 17 – Вкладка «Главная» таблицы «UDO\_T\_ORF\_PAY»

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Storage	Cor
RN	<input type="checkbox"/>	NUMBER(17)	<input type="checkbox"/>			<input type="checkbox"/>		
COMPANY	<input type="checkbox"/>	NUMBER(17)	<input type="checkbox"/>			<input type="checkbox"/>		
CRN	<input type="checkbox"/>	NUMBER(17)	<input type="checkbox"/>			<input type="checkbox"/>		
PRN	<input type="checkbox"/>	NUMBER(17)	<input type="checkbox"/>			<input type="checkbox"/>		
YEAR	<input type="checkbox"/>	NUMBER(4)	<input type="checkbox"/>			<input type="checkbox"/>		
MONTH	<input type="checkbox"/>	NUMBER(2)	<input type="checkbox"/>	1		<input type="checkbox"/>		
SUMM	<input type="checkbox"/>	NUMBER(17,2)	<input type="checkbox"/>	0		<input type="checkbox"/>		
PAYDATE	<input type="checkbox"/>	DATE	<input type="checkbox"/>			<input type="checkbox"/>		
PAYTYPE	<input type="checkbox"/>	NUMBER(1)	<input type="checkbox"/>	0		<input type="checkbox"/>		
FICTION	<input type="checkbox"/>	NUMBER(1)	<input type="checkbox"/>	0		<input type="checkbox"/>		
DOCTYPE	<input type="checkbox"/>	NUMBER(17)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
DOCNUMB	<input type="checkbox"/>	VARCHAR2(240)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
DOCDATE	<input type="checkbox"/>	DATE	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
NOTE	<input type="checkbox"/>	VARCHAR2(2000)	<input checked="" type="checkbox"/>			<input type="checkbox"/>		
AVAILABLE	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	"PARUS"."UDO_F_ORF_TRUE"("RN","COMPANY","UDO_ORF_PAY")		<input type="checkbox"/>		
UNLOADED	<input type="checkbox"/>	NUMBER(1)	<input checked="" type="checkbox"/>	0		<input type="checkbox"/>		0 - 4

Рисунок 18 – Вкладка «Поля» таблицы «UDO\_T\_ORF\_PAY»

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated
UDO_ORF_PAY_PR	Primary	RN	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UDO_ORF_PAY_COMPANY	Foreign	COMPANY	<input checked="" type="checkbox"/>	COMPANIES	RN	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UDO_ORF_PAY_CRN	Foreign	CRN	<input checked="" type="checkbox"/>	ACATALOG	RN	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UDO_ORF_PAY_DOCTYPE	Foreign	DOCTYPE	<input checked="" type="checkbox"/>	DOCTYPES	RN	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UDO_ORF_PAY_PRN	Foreign	PRN	<input checked="" type="checkbox"/>	UDO_T_ORDERS_FOOD	RN	Cascade	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 19 – Вкладка «Ключи» таблицы «UDO\_T\_ORF\_PAY»

Name	Condition	Enabled	Deferrable	Deferred	Validated	Last change
C_UDO_ORF_PAY_DOCNUMB	DOCNUMB is null or rtrim(DOCNUMB) is not null	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	01.12.2017 18:08:51
C_UDO_ORF_PAY_FICTION	FICTION in (0,1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	01.12.2017 18:08:51
C_UDO_ORF_PAY_PAYTYPE	PAYTYPE in (0,1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	01.12.2017 18:08:51

Рисунок 20 – Вкладка «Проверки» таблицы «UDO\_T\_ORF\_PAY»

Owner	Name	Type	Columns	Compress	Logging	Prefix length	Invisible	Local	Reverse	Storage
PARUS	I_UDO_ORF_PAY_CRN_FK	Normal	CRN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfre
PARUS	I_UDO_ORF_PAY_DOCTYPE_FK	Normal	DOCTYPE	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfre
PARUS	I_UDO_ORF_PAY_PRN_FK	Normal	PRN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfre
PARUS	UDO_ORF_PAY_PR	Unique	RN	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	tablespace parus_main pctfre

Рисунок 21 – Вкладка «Индексы» таблицы «UDO\_T\_ORF\_PAY»

Таким образом был разработан раздел «Заказы питания». Внешний вид раздела представлен на рисунках 22 – 25.

ПАРУС-Бюджет 8 Расчет оплаты услуг учреждения - [Заказы питания]

Каталоги

- Заказы питания
  - Беловское СШ
  - Беломестное СШ
  - Бессоновка СШ
  - Ближн. Игум. СШ
  - Вес.Лопань. СШ
  - ГоловиноСШ
  - Дубовое СОШ
  - Дубовская НШ
  - Ерик ОС
  - Журавлевка СШ
  - Комсомolec СШ
  - Крас.Октябрь СШ
  - Красный Хут. ОШ
  - Крут.Лог СШ
  - Майская гимназия
  - Мясоедово СШ

Заказы питания (Объём выборки: 387)

<input checked="" type="checkbox"/>	Период	Префикс	Номер лич	Учреждение	Номер лицевого счёта	Региональный бюджет	Муниципальный бюджет	Родительская плат
<input type="checkbox"/>	1/2018	2017	15221	Бессоновка СШ	05496477	0,00	32,50	30,00
<input type="checkbox"/>	1/2018	2017	15222	Бессоновка СШ	05496483	0,00	32,50	30,00
<input type="checkbox"/>	1/2018	2017	15223	Бессоновка СШ	05496497	0,00	32,50	30,00

Заказы питания (Питание) | Заказы питания (Оплаты) | Заказы питания (Начисления) | Заказы питания (Расчет за месяц)

Заказы питания (Питание)

<input checked="" type="checkbox"/>	Дата	Наименование блюда	Тип питания	Региональный бюджет	Муниципальный бюджет	Родительская плата	Статус
<input type="checkbox"/>	09.01.2018	Горячий молочный завтрак дл	Завтрак	0,00	32,50	0,00	Не выгружено
<input type="checkbox"/>	09.01.2018	Горячий обед для всех катег	Обед	0,00	0,00	30,00	Не выгружено

Рисунок 22 – Раздел «Заказы питания»

Раздел «Заказы питания» содержит 4 спецификации: «Питание», «Начисления», «Оплаты» и «Расчет за месяц».

Период	Префикс	Номер лич.	Учреждение	Номер лицевого счёта	Региональный бюджет	Муниципальный
1/2018	2017	15221	Бессоновка СШ	05496477	0,00	0,00
1/2018	2017	15222	Бессоновка СШ	05496483	0,00	0,00
1/2018	2017	15223	Бессоновка СШ	05496497	0,00	0,00

Период "В" (месяц)	Период "В" (год)	Сумма	Дата платёж	Тип оплаты	Погашение задолжен.
Январь	2018	450,00	15.01.2018	Оплата	Нет

Рисунок 23 – Спецификация «Оплаты»

Спецификация «Оплаты» отображает данные о поступивших денежных средствах. Реестры платежей от Сбербанка загружаются в раздел «Реестры на зачисления» после чего обрабатываются в раздел «Заказы питания».

Наименование	Дата	Год	Количество	Региональный бюджет	Муниципальный бюджет	Родительская оплата	Ф.И.О.
Сырники молочный завтрак д/л	Январь	2018	1	0,00	32,50	30,00	Баранов
Горячий обед для всех катег.	Январь	2018	1	0,00	0,00	30,00	Бондарь

Рисунок 24 – Спецификация «Начисления»

В спецификации «Начисления» суммарно отображаются заказы питания за текущий месяц.

Тип	Месяц	Год	Сумма
Входящий остаток	Январь	2018	0,00
Начислено	Январь	2018	30,00
Оплачено	Январь	2018	450,00
Возвращено	Январь	2018	0,00
Исходящий остаток	Январь	2018	-420,00

Рисунок 25 – Спецификация «Расчет за месяц»

Спецификация «Расчет за месяц» наглядно отображает информацию о начислениях, поступивших платежах и об остатках за месяц.

Разработка раздела «Журнал взаимодействия» производится также в модуле «Администратор» ПП «Парус 8». Структура раздела представлена на рисунке 26.

Код раздела	Имя таблицы	Наименование раздела	Деление по каталогу	Деление по юридичес	Использование свойств	Абстрактный раздел	Пользовательский	Иерархич
BELVSOPENJRSP	BELVSOPENJRSP	Журнал взаимодействия с ИСОУ (виртуальная школа (специализация))	да	нет	нет	нет	Конструктор	нет

Тип	Код	Наименование	Позиция	Пользовательское действие	Режим обработки з	Режим завершения тр.	Режим обнов
Стандартное	BELVSOPENJRSP_VIEWMESSAGE	Промотр сообщения		Конструктор	Для одной таблицы з	После каждого вызова	не обновля

Рисунок 26 – Структура раздела «Журнал взаимодействия»

Для раздела «Журнал взаимодействия» необходимо создать таблицу данных «UDO\_T\_ORF\_MONTH\_CALC».

Создание таблицы данных «UDO\_T\_ORF\_MONTH\_CALC» представлено на рисунках 27 – 31.

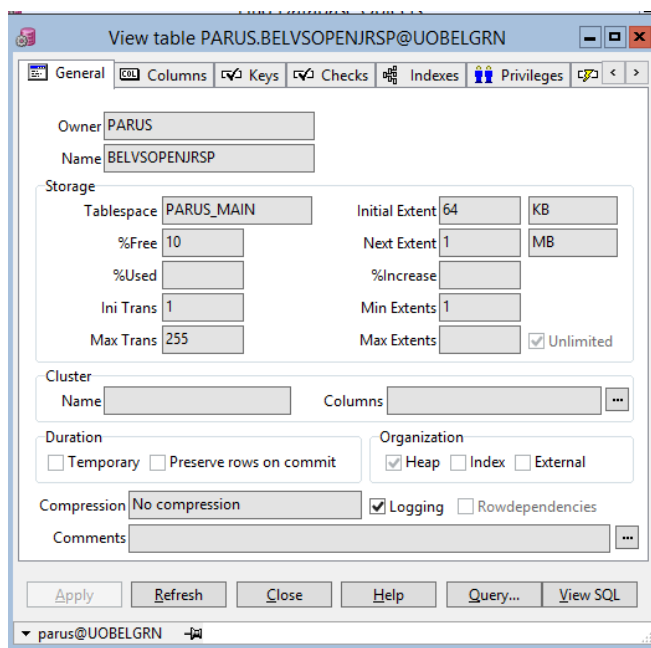


Рисунок 27 – Вкладка «Главная» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

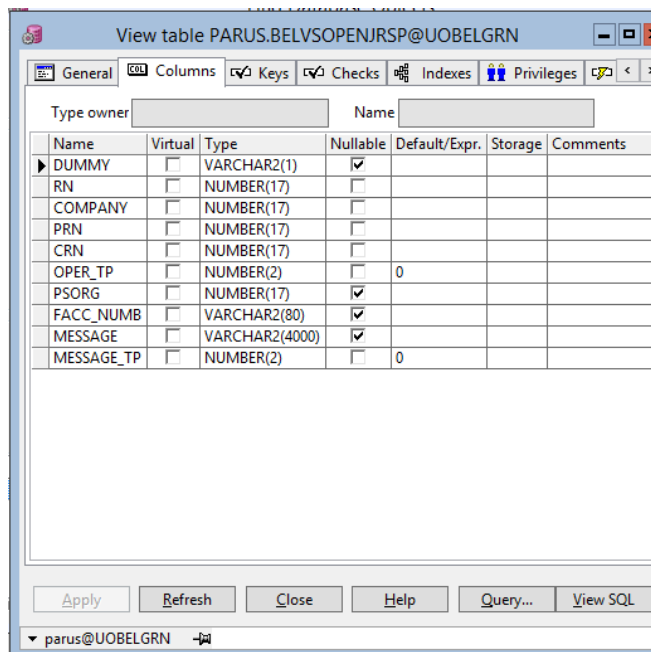


Рисунок 28 – Вкладка «Колонки» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

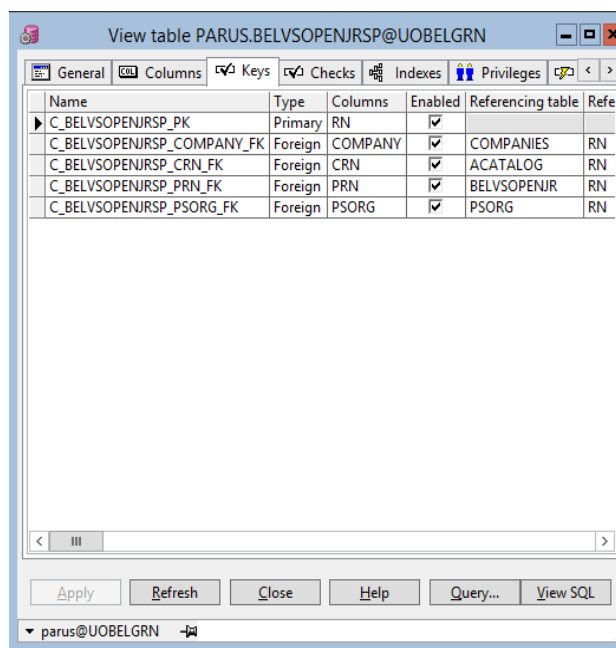


Рисунок 29 – Вкладка «Ключи» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

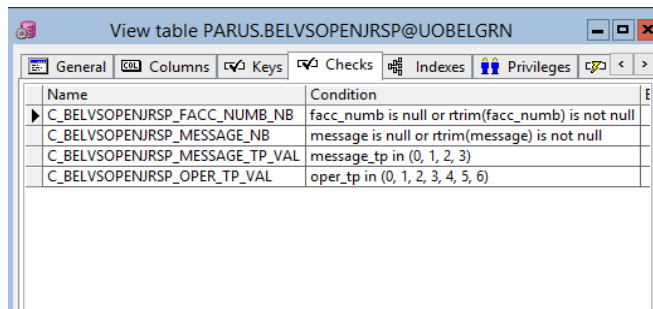


Рисунок 30 – Вкладка «Отметка» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

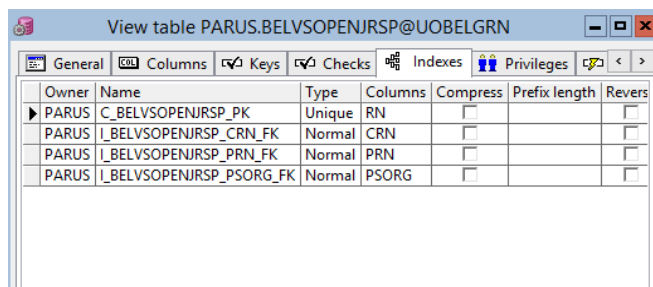


Рисунок 31 – Вкладка «Индексы» таблицы «UDO\_T\_ORF\_MONTH\_CALC»

Таким образом был разработан раздел «Журнал взаимодействия». Внешний вид раздела представлен на рисунках 32 - 34.

Журнал взаимодействия с ИСОУ Виртуальная школа							
Пользователь	Вид операции	Идентификатор доку	Дата начала действ	Дата окончания действ	Количество сообщени	Количество ошибок	Время выполнения
<input type="checkbox"/> PARUS	Получение данных об учениках	1481	16.03.2018 16:05:25	16.03.2018 16:05:34	0	0	00:00:09
<input type="checkbox"/> PARUS	Получение данных об учениках	1542	20.03.2018 11:56:50	20.03.2018 11:57:03	0	0	00:00:13
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1911	03.04.2018 14:12:54	03.04.2018 14:38:12	0	0	00:25:18
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1909	03.04.2018 13:20:35	03.04.2018 13:39:49	0	0	00:19:14
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1904	03.04.2018 11:38:52	03.04.2018 11:53:01	0	0	00:14:09
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1902	03.04.2018 11:18:26	03.04.2018 11:38:08	0	0	00:19:42
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1912	03.04.2018 14:47:44	03.04.2018 15:10:35	0	0	00:22:51
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1910	03.04.2018 13:50:48	03.04.2018 14:10:48	0	0	00:20:00
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1901	03.04.2018 10:59:15	03.04.2018 11:17:18	0	0	00:18:03
<input type="checkbox"/> PARUS	Получение данных о заказах питания	1899	03.04.2018 10:23:46	03.04.2018 10:38:20	0	0	00:14:34

Рисунок 32 – Заголовок раздела «Журнал взаимодействия»

Журнал взаимодействия с ИСОУ Виртуальная школа (спецификация)				
Текст сообщения	Вид операции	Идентификатор доку	Текст сообщения	
<input type="checkbox"/> Ошибка	Получение данных о заказах питания	58449	12722782	Не удалось добавить заказ питания. Школа: 58449 Класс: 437999 Дата: 29.01.2018 08:04:05.12: в "PARUS.UDO_F_ORDERS_FOOD_BASE_INSERT", line 24 ORA-06512: в "PARUS.PRO_BELVSOPENJR_CORE", line 426
<input type="checkbox"/> Ошибка	Получение данных о заказах питания	58449	12722782	Не удалось добавить заказ питания. Школа: 58449 Класс: 437999 Дата: 29.01.2018 08:04:05.12: в "PARUS.UDO_F_ORDERS_FOOD_BASE_INSERT", line 14 ORA-06512: в "PARUS.PRO_BELVSOPENJR_CORE", line 426

Рисунок 33 – Спецификация раздела «Журнал взаимодействия»



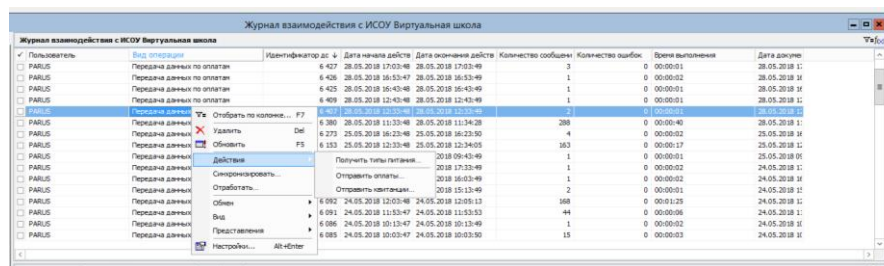


Рисунок 34 – Действия в разделе «Журнал взаимодействия»

### 3.2 Разработка web-API

Создание web-сервиса API производится с помощью программного обеспечения Visual Studio компании Microsoft.

Visual Studio – интегрированная среда разработки программного обеспечения, позволяющая создавать, как консольные приложения, так и приложения с графическим интерфейсом.

Для решения задачи, связанной с разработкой интеграции необходимо выполнить следующее: [19;20]

- Создать новый проект в Visual Studio. Для этого выполнить действие File - New – Project.
- Далее необходимо выбрать шаблон ASP.NET Web Application.

Проект интеграции систем назовём «VirtualSchool». Рисунок 35.

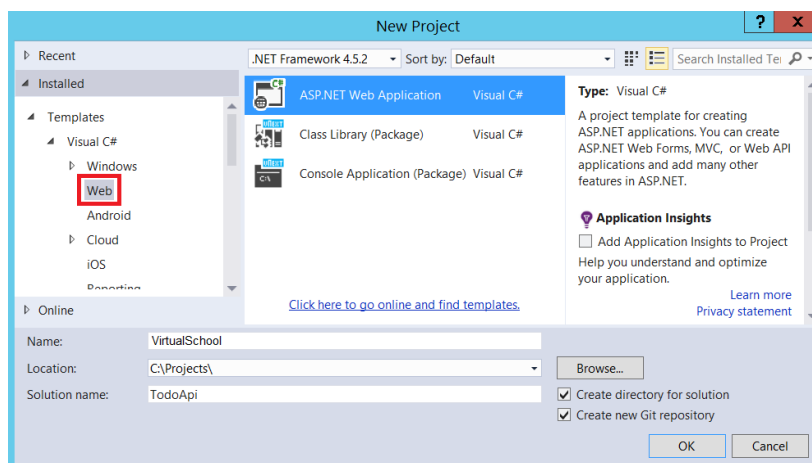


Рисунок 35 – Создание шаблона ASP.NET Web Application

– Далее в окне Новый проект выбираем Web API для ASP.NET 5 Preview Templates. Рисунок 36.

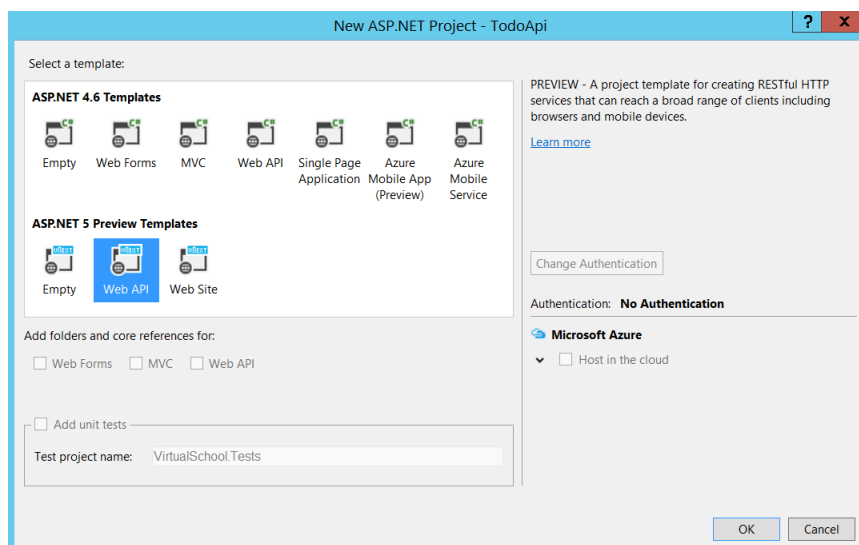


Рисунок 36 – Создание проекта Web API

– Далее необходимо добавить модельные классы, для представления данных в системе интеграции. Для этого необходимо по модели VirtualScholl кликнуть правой кнопкой мыши и выбрать Add - New Item. В открывшемся окне выбрать шаблон Class [20].

– После чего, необходимо добавить классы репозитория. Класс репозитория необходим для хранения данных и логики их получения и отправки.

– Для внедрения репозитория в контроллер, необходимо зарегистрировать репозиторий с помощью DI-контейнера.

– Для получения и/или отправки данных от сервиса ИСОУ «Виртуальная школа» необходимо создать соответствующие методы (GET, PUT, UPDATE,) в классе контроллера. Методы описаны в приложении Е. При обращении к сервису, данные будут возвращаться в зависимости от типа файла в описанном формате json. [23].

– Для проверки подлинности и авторизации используем функцию OAuth, она позволяет хранить учетные данные клиента и отправлять маркеры. В OAuth используется переменная «issuer», которая является

уникальным идентификатором для сущности, также используется переменная «secret» - секретный ключ, используемый для маркера безопасности.

– Для загрузки данных в ПП «Парус 8» необходимо написать запрос для обращения на web-сервис, см. приложение Е [25].

В данном разделе были разработаны разделы ПП «Парус 8» и web-API для интеграции информационной системы образовательных учреждений «Виртуальная школа» и программного продукта «Парус 8». Разработанная интеграция прошла успешное тестирование и была внедрена в деятельность компании «Парусник-Белгород». Положительный результат тестирования показывает успешное выполнение поставленных задач и как следствие достижение поставленной цели. Разработанная интеграция информационных систем обеспечивает ведение учета оплаты питания.

## ЗАКЛЮЧЕНИЕ

Интеграция информационных систем - это взаимодействие разных систем, их блоков, передача данных в разных форматах. От того, как налажен обмен данными, зависит эффективность работы систем, целостность и непротиворечивость передаваемой информации. Данный фактор свидетельствует о том, что разработанная система интеграции информационной системы образовательных учреждений «Виртуальная школа» и программного продукта «Парус 8» является актуальной.

Цель магистерской диссертации заключалась в совершенствовании учета оплаты школьного питания за счет его автоматизации путем интеграции ИСОУ «Виртуальная школа» и ПП «Парус 8».

Для достижения поставленной цели, в работе были выполнены следующие задачи:

- провели анализ существующих методов и средств интеграции специализированных ИС;
- выбрали метод реализации интеграции ИС;
- спроектировали интеграцию ИС;
- разработали интеграцию ИС.

Разработанная интеграция ИС обеспечивает:

- обмен данными об учащихся в образовательных учреждениях;
- обмен данными по заказам питания в образовательных учреждениях;
- обмен данными по поступившим оплатам от родителей;
- обмен данными по состоянию лицевого счета ребенка;
- передачу квитанций для оплаты школьного питания.

В дальнейшем разработанную интеграцию можно расширить путем описания новых форматов для обработки и передачи информации в рамках взаимодействия ИСОУ «Виртуальная школа» и ПП «Парус 8».

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 Степанов, Д.Ю. Перспективные направления развития корпоративных информационных систем на примере программных решений компании SAP / Д. Ю. Степанова // Аспирант и соискатель. – 2013. – Т.66.
- 2 Лодон, Дж. Управление информационными системами./ Дж. Лодон, К. Лодон. - Пер. с англ. под ред. Трутнева Д.Р. – СПб.: Питер. – 2005.
- 3 Кусов, А.А. Проблемы интеграции корпоративных информационных систем / А.А. Кусов // Управление экономическими системами: электронный научный журнал. – 2011. – Т. 28, №4.
- 4 Anica, P. A Framework for Enhancing Competitive Intelligence Capabilities using Decision Support System based on Web Mining Techniques. / P. Anica // Int. J. of Computers, Communications & Control. – 2009.
- 5 Guido, A.L. Semantic Integration of Information Systems. / A.L. Guido // International Journal of Computer Networks & Communications. – 2011.
- 6 Hohhof, B. Developing Information Systems for Competitive Intelligence Support. B. Hohhof // Library Trend. – 2012.
- 7 Дли, М. И. Способы интеграции информационных систем субъектов экономической деятельности при использовании аутсорсинга. М.И. Дли // Национальный исследовательский университет «МЭИ», г. Смоленск. – 2017.
- 8 Морозова, О.А. Интеграция корпоративных информационных систем. / О.А. Морозова // М.: Финансовый университет, – 2014. – 140 с.
- 9 Enterprise Connectivity Patterns: Implementing integration solutions with IBM. Электронный ресурс // Режим доступа: [http://www.ibm.com/developerworks/webservices/library/ws-enterpriseconnectivitypatterns/index.html?S\\_TACT=105AGX99&S\\_CMP=CP](http://www.ibm.com/developerworks/webservices/library/ws-enterpriseconnectivitypatterns/index.html?S_TACT=105AGX99&S_CMP=CP).
- 10 Gamma et al. Design Pattern – Elements of Reusable Object Orientated Software. / Addison-Wesley. – 2001.
- 11 Integration Patterns Overview. Электронный ресурс // Режим доступа: <http://www.enterpriseintegrationpatterns.com/eaipatterns.html>.

12 Microsoft Message Queuing (MSMQ) - промежуточная среда обмена сообщениями // Microsoft. - Режим доступа: <http://www.intuit.ru/department/se/msfdev/6/1.html>.

13 OASIS Web Services Business Process Execution Language (WSBPEL) TC // Режим доступа: [www.oasisopen.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=wsbpel).

14 Артамонов, И. Современные стандарты описания и исполнения бизнес-процессов. Электронный ресурс // Режим доступа: URL: <http://esm-journal.ru/post/Sovremennye-standarty-opisanija-i-ispolnenija-biznes-processov.aspx>.

15 Бин, Д. XML для проектировщиков. Повторное использование и интеграция /Д. Бин // М.: КУДИНЦ-ОБРАЗ. – 2004.

16 Волкова, И.А. Системы программирования: учебное пособие /Д. Бин // М.: Издательский отдел факультета ВМК МГУ. – 2009.

17 Гандерлой, М. Освоение Microsoft SQL Server: пер с англ / М. Гандерлой // М.: ИД «Вильямс». – 2009.

18 Горин, С.В. Поддержка разработки распределенных приложений в Microsoft .NET Framework / С.В. Горин // М.: МГТУим. Баумана. – 2006.

19 Зайден, М. XML для электронной коммерции /М. Зайден// М.: Бином: Лаборатория знаний. – 2010.

20 Разработка Web-сервисов в XML и серверных компонентов на Visual Basic .NET и Visual C# .NET. / М.: Русская. –2011.

21 Руководство Microsoft по проектированию архитектуры приложений / Электронный ресурс // Корпорация «Майкрософт». – 2009. Режим доступа: <https://www.microsoft.com/en-us/download>

22 Самуйлов, К.Е. Бизнес-процессы и информационные технологии в управлении телекоммуникационными компаниями /К.Е. Самуйлов // М.: Альпина Паблишерз. – 2009.

23 Хабибулин, И.Ш. Самоучитель XML /И.Ш. Хабибулин // СПб.: БХВ-Петербург. – 2003.

24 Хоп, Г. Шаблоны интеграции корпоративных приложений: пер с англ / Г. Хоп // М.: ИД «Вильямс». – 2007.

25 Хохгуртль, Б. С# и Java: межплатформенные Web-сервисы /Б. Хохгуртль // М.: КУДИНЦ-ОБРАЗ. – 2013.

26 Эммерих, В. Конструирование распределенных объектов. Методы и средства программирования интерпортабельных объектов в архитектурах OMG/CORBA, Microsoft COM и Java RMI: пер.с англ / В. Эммерих // М.: Мир. – 2012.

## ПРИЛОЖЕНИЯ



## ПРИЛОЖЕНИЕ А

Техническое задание на разработку системы интеграции специализированных информационных систем ПП «Парус 8» и ИСОУ «Виртуальная школа»

### 1. Общие сведения

#### 1.1 Наименование системы

##### 1.1.1 Полное наименование системы

Интеграция специализированных информационных систем ПП «Парус 8» и ИСОУ «Виртуальная школа».

##### 1.1.2 Краткое наименование системы

Система, интеграция.

#### 1.2 Основания для проведения работ

Работа выполняется в рамках трудового договора между ООО «Парусник-Белгород» и Дегтяревой А.М.

#### 1.3 Наименование организаций – Заказчика и Разработчика

##### 1.3.1 Заказчик

Заказчик: ООО «Парусник - Белгород»

Адрес: г. Белгород, Белгородский проспект 77

##### 1.3.2 Разработчик

Разработчик: Дегтярева Анастасия Михайловна – экспер-аналитик  
ООО «Парусник-Белгород»

Адрес: г. Белгород, Белгородский проспект 77

#### 1.4 Плановые сроки начала и окончания работы

Начало работ – сентябрь 2017

Окончание работ – май 2018

#### 1.5 Источники и порядок финансирования

Финансирование осуществляется в рамках трудового договора между ООО «Парусник-Белгород» и Дегтяревой А.М.

## 1.6 Порядок оформления и предъявления заказчику результатов работ

Работы по созданию системы интеграции сдаются разработчиком поэтапно в соответствии с планом-графиком. По окончании выполнения каждого из этапов производится внедрение системы интеграции.

## 2. Назначение и цели создания системы

### 2.1 Назначение системы

Интеграция специализированных систем ПП «Парус 8» и ИСОУ «Виртуальная школа» предназначена для бухгалтерского учета безналичной оплаты школьного питания.

### 2.2 Цели создания системы

Интеграция реализуется с целью перехода на безналичную оплату школьного питания в рамках выполнения губернаторской программы «Безналичный мир Белогорья».

В результате создания интеграции систем:

- нет движения наличных денежных средств между детьми и учителями;
- для родителей доступна информация по питанию ребенка за любой период времени;
- возможна безналичная оплата питания;
- доступна информация по движению и остаткам денежных средств в разрезе лицевого счета в бухгалтерском учете.

## 3. Характеристика объектов автоматизации

Компания «Парусник-Белгород» является генеральным дилером корпорации «Парус». Основным видом деятельности является комплексная автоматизация бухгалтерского учета в бюджетных учреждениях. Также, компания «Парусник-Белгород» занимается автоматизацией кадрового учета, финансового планирования и электронным документооборотом.

Компания «Парусник - Белгород» имеет следующую организационную структуру:

Руководство компании; Бухгалтерия; Управление анализа и разработки; Отделы обслуживания клиентов; Отдел продаж и корпоративного развития; Отдел договоров и товарной логистики.

Разработка интеграции специализированных информационных систем проводится совместно с Управлением анализа и разработки.

#### 4. Требования к системе

Для интеграции систем в ПП «Парус 8» должно быть реализовано следующее:

- Разработан раздел «Заказы питания», который должен отражать информацию из штатного раздела «Личные карточки» в частности ФИО, лицевой счет, дату начала и дату конца действия, подведомственное учреждение и класс, в котором числится ученик. Также, раздел «Заказы питания» должен хранить информацию по типам питания, начислению и списанию денежных средств и оплатах.

- В раздел «Заказы питания» должна осуществляться загрузка данных от «ИСОУ «Виртуальная школа» и выгрузка данных для «ИСОУ «Виртуальная школа» и банка.

- Загрузка данных по учащимся: персональные данные, принадлежность к учреждению и состоянию лицевого счета.

- Выгрузка квитанций по лицевым счетам.

- Выгрузка данных по состоянию баланса лицевого счета.

- Разработан раздел «Журнал взаимодействия» для хранения и обработки данных в результате интеграции с «ИСОУ «Виртуальная школа» с помощью API сервиса.

- Разработана интеграция специализированных информационных системам «Парус 8» и «ИСОУ «Виртуальная школа» с помощью веб-сервиса по средствам API.

## 5. Состав и содержание работ по созданию системы

Работы по созданию интеграции должны выполняться по следующему графику:

Сентябрь 2017 – Октябрь 2017 – разработка раздела «Заказы питания», написание загрузки данных от «ИСОУ «Виртуальная школа».

Ноябрь 2017 – Январь 2018 – разработка раздела «Типы питания» и «Журнал взаимодействия».

Февраль 2018 – Апрель 2018 – разработки выгрузки данных из ПП «Парус 8». Настройка интеграции через веб-сервис.

Май 2018 – Полное внедрение системы интеграции, устранение ошибок на этапе внедрение. Передача функционала клиентам компании «Парусник - Белгород». Написание технической документации.

Заказчик:

ООО «Парусник - Белгород»

Ген. Директор \_\_\_\_\_ Максимов А.А.

Исполнитель:

\_\_\_\_\_ Дегтярева А.М.

## ПРИЛОЖЕНИЕ Б

### Тело запроса объектов в json для заказов питания

```
{
  "title": "Food Orders",
  "description": "JSON schema for Food Orders",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "SchoolId": {
        "type": "integer"
      },
      "ClassId": {
        "type": "integer"
      },
      "CheckingAccount": {
        "type": "string",
        "maxLength": 10
      },
      "DateTime": {
        "type": "string",
        "format": "date-time"
      },
      "StudentFullName": {
        "type": "string",
        "maxLength": 770
      },
      "FoodTypeId": {
        "type": "integer"
      },
      "RegionalBudget": {
        "type": "integer"
      },
      "MunicipalBudget": {
        "type": "integer"
      },
      "ParentalPayment": {
        "type": "integer"
      }
    }
  }
}
```

## ПРИЛОЖЕНИЕ В

### Тело запроса объектов в json для учеников

```
{
  "title": "Students",
  "description": "JSON schema for Students",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "SchoolId": {
        "type": "integer"
      },
      "SchoolName": {
        "type": "string",
        "maxLength": 256
      },
      "ClassId": {
        "type": "integer"
      },
      "ClassName": {
        "type": "string",
        "maxLength": 256
      },
      "CheckingAccount": {
        "type": "string",
        "maxLength": 10
      },
      "StudentFullName": {
        "type": "string",
        "maxLength": 770
      },
      "RecommendedPayment": {
        "type": "number"
      },
      "StudentActive": {
        "type": "boolean"
      },
      "EnrollmentDate": {
        "type": "string",
        "format": "date-time"
      },
      "DismissalDate": {
        "type": "string",
        "format": "date-time"
      }
    }
  }
}
```

## ПРИЛОЖЕНИЕ Г

### Тело запроса объектов в json для квитанций

```
{
  "title": "Receipts",
  "description": "JSON schema for Receipts",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "Payee": {
        "type": "string"
      },
      "PayeePersonalAccount": {
        "type": "string"
      },
      "PayeeInn": {
        "type": "string"
      },
      "Oktmo": {
        "type": "string"
      },
      "BankName": {
        "type": "string"
      },
      "Bic": {
        "type": "string"
      },
      "CheckingAccount": {
        "type": "string"
      },
      "Cbc": {
        "type": "string"
      },
      "Purpose": {
        "type": "string"
      },
      "ChildFullName": {
        "type": "string"
      },
      "ChildPersonalAccount": {
        "type": "string"
      },
      "PaymentType": {
        "type": "string"
      },
      "ClassNum": {
        "type": "string"
      },
      "PaymentPeriod": {
        "type": "string"
      },
      "Sum": {
        "type": "number"
      }
    }
  }
}
```

## ПРИЛОЖЕНИЕ Д

### Тело запроса объектов в json для оплат

```
{
  "title": "Payments",
  "description": "JSON schema for Payments",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "CheckingAccount": {
        "type": "string",
        "maxLength": 10
      },
      "DateTime": {
        "type": "string",
        "format": "date-time"
      },
      "BalanceBefore": {
        "type": "number"
      },
      "Incoming": {
        "type": "number"
      },
      "WriteOff": {
        "type": "number"
      },
      "Balance": {
        "type": "number"
      }
    }
  }
}
```



## ПРИЛОЖЕНИЕ Е

### Запрос для обращения на web-сервис

```
create or replace procedure bel_p_food_types_load_api(p_company in number)
as
begin
    /* Loading food types */
    bel_pkg_virtualscholl_api.loadFoodTypes(p_company                => p_company,
                                             p_food_types_endpoint =>
'/api/v1/food_types');
end;
```

```
create or replace procedure bel_p_food_orders_load_api(p_company in number)
as
begin
    /* Loading food orders */
    bel_pkg_virtualscholl_api.loadFoodOrders(p_company                =>
p_company,
                                             p_food_orders_endpoint =>
'/api/v1/food_orders/' || Bel_TDateTimeOffset.toUnixTimeMilliseconds());
end;
```

```
create or replace procedure bel_p_students_load_api(p_company in number)
as
begin
    /* Loading students */
    bel_pkg_virtualscholl_api.loadStudents(p_company                => p_company,
                                             p_students_endpoint => '/api/v1/students/'
|| Bel_TDateTimeOffset.toUnixTimeMilliseconds());
end;
```

```
create or replace procedure bel_p_payments_unload_api(p_company in number)
as
begin
    /* Unload payments */
    bel_pkg_virtualscholl_api.unloadPayments(p_company                => p_company,
                                             p_payments_endpoint =>
'/api/v1/payments');
end;
```

Магистерская диссертация выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«      » \_\_\_\_\_

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(Ф.И.О.)