

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

«Прикладная информатика в социальной сфере»

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Автоматизация процесса организации цепей поставок
производственного предприятия (на примере ООО «ВСП»)

Студент

М.А. Прохорова

(И.О. Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

Бакалаврская работа посвящена актуальной тематике автоматизации процесса по организации цепей поставок производственного предприятия. В рамках работы были выявлены недостатки в процессе выбора транспортной компании для перевозки грузов; обоснован выбор информационной системы для автоматизации процесса; разработан программный продукт для автоматизации процесса по организации цепей поставок производственного предприятия, проведен расчёт экономической эффективности внедрения информационной системы.

Цель выпускной квалификационной работы – разработка информационной системы для автоматизации процесса по организации цепей поставок производственного предприятия ООО «ВСП».

Объект исследования выпускной квалификационной работы – процесс заказа авто компонентов в ООО «ВСП».

Предмет исследования выпускной квалификационной работы – автоматизация процесса заказа авто компонентов в ООО «ВСП».

Метод решения задачи – бизнес-моделирование информационных систем.

В заключительной части был произведен расчёт экономической эффективности проекта. Эффективность внедрения проекта обуславливается следующими показателями: окупаемость проекта составила 2,5 месяца, трудоемкость процесса снижается на 6% а стоимость транспортировки снижается на 30%.

Оглавление

Введение.....	5
Глава 1 Анализ процесса заказа авто компонентов в ООО «ВСП»	8
1.1 Характеристика производственного предприятия.....	8
1.1.1 Характеристика процесса заказа производственных компонентов....	9
1.2 Концептуальное моделирование информационной системы по автоматизации процесса по организации цепей поставок производственного предприятия.....	13
1.2.1 Выбор методологии и технологии концептуального моделирования информационной системы	13
1.2.2 Моделирование бизнес-процесса «КАК ЕСТЬ» текущего процесса по заказу компонентов.	14
1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии	17
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	19
1.3.1 Определение критериев анализа	19
1.3.2 Сравнительная характеристика существующих разработок.....	22
1.4 Постановка задачи на разработку проекта создания/внедрения информационной системы автоматизации процесса по организации цепей поставок.....	23
1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»	24
Глава 2 Логическое проектирование информационной системы автоматизации процесса организации цепей поставок производственного предприятия	27
2.1 Выбор технологии логического моделирования информационной системы	27
2.1.1 Разработка требований к информационной системе	27

2.1.2	Разработка диаграммы классов информационной системы управления цепями поставок производственного предприятия.....	29
2.1.3	Разработка диаграммы последовательностей управления цепями поставок производственного предприятия	30
2.2	Информационное обеспечение информационной системы	33
2.2.1	Используемые классификаторы и системы кодирования	33
2.2.2	Характеристика нормативно-справочной и входной оперативной информации	34
2.2.3	Характеристика выходной информации	35
2.3	Проектирование базы данных информационной системы автоматизации процесса организации цепей поставок производственного предприятия. ..	37
2.3.1	Выбор технологии проектирования базы данных информационной системы автоматизации процесса организации цепей поставок производственного предприятия.....	37
2.3.2	Разработка концептуальной модели данных информационной системы	37
2.3.3	Обоснование вида логической модели	39
2.3.4	Разработка логической модели данных информационной системы	40
2.4	Требования к аппаратно-программному обеспечению информационной системы управления цепями поставок производственного предприятия....	42
Глава 3. Физическое проектирование информационной системы управления цепями поставок		44
3.1	Выбор архитектуры информационной системы управления цепями поставок.....	44
3.2	Выбор технологии разработки программного обеспечения информационной системы управления цепями поставок	46
3.3	Выбор системы управления базами данных информационной системы управления цепями поставок	49
3.4	Разработка физической модели данных информационной системы управления цепями поставок	50

3.5 Разработка программного обеспечения информационной системы управления цепями поставок	51
3.5.1 Схема взаимосвязи модулей приложения информационной системы управления цепями поставок	51
3.5.2 Описание модулей приложения информационной системы управления цепями поставок с примерами программного кода.....	53
3.6 Описание функциональности информационной системы управления цепями поставок.....	56
3.7 Оценка и обоснование экономической эффективности разработки информационной системы автоматизации процесса по организации цепей поставок.....	58
3.7.1 Выбор методики расчета экономической эффективности	58
3.7.2 Расчет показателей экономической эффективности проекта	59
Заключение	63
Список используемой литературы и используемых источников.....	65
Приложение А Исходный код программы.....	67

Введение

Процесс заказа компонентов на производственном предприятии - это система, связанная с обработкой и хранением большого количества потоков информации из разных источников. Заказ авто компонентов относится к процессу планирования производства: отправка прогноза поставщикам, отправка заказа поставщикам, процесс выбора перевозчика, процесс транспортировки и прохождения таможенной проверки, приёмка компонентов на производственном предприятии.

В вышеуказанном процессе формируется огромное количество документов, необходимых для отправки, таможни, приёмки. В существующем процессе много времени уходит на подготовку и проверку заполнения документов. Так как все документы готовятся работниками нельзя исключать человеческий фактор, который может негативно влиять на успешную поставку.

Ошибочные прогнозы и заказы вызывают задержки в доставке и дополнительные расходы, недовольство потребителей и производства, сбой в работе производства, остановка производственных линий потребителя, сокращения продаж предприятия и незапланированные издержки.

В производственном предприятии ООО «ВСП» (г. Тольятти) для повышения точности работы планирования и более эффективной работы системы было принято решение по внедрению информационной системы автоматизации процесса по организации цепей поставок производственного предприятия, разработанной на основе современных IT-технологий.

Таким образом, актуальность темы выпускной квалификационной работы обусловлена необходимостью автоматизации процесса по организации цепей поставок в ООО «ВСП».

Целью работы является разработка информационной системы для автоматизации процесса по организации цепей поставок производственного предприятия ООО «ВСП».

Объектом исследования выпускной квалификационной работы – процесс заказа авто компонентов в ООО «ВСП»

Предмет исследования – автоматизация процесса заказа авто компонентов в ООО «ВСП»

Для достижения цели выпускной квалификационной работы необходимо решить следующие задачи:

- сделать анализ существующего процесса заказа компонентов, цепь поставок и планирования, оценить недостатки и разработать решение по совершенствованию процесса с использованием современных технологий;
- разработать концептуальную модель информационной системы автоматизации процесса по организации цепей поставок и сделать анализ существующих IT-решений, предлагаемым на рынке на предмет соответствия необходимым требованиям;
- разработать логическую модель информационной системы;
- создать программный продукт для реализации информационной системы.

Для решения поставленных задач были использованы следующие методы исследования: эмпирический сбор данных о среде реализации, эмпирическое моделирование процесса, проектирование информационной системы.

Выпускная квалификационная работа состоит из введения, двух глав, заключения и приложения.

Во введении обосновывается актуальность темы, определяются объект, предмет и цель ВКР, перечисляются задачи, которые необходимо решить для ее достижения.

В первой главе дана характеристика производственного предприятия ООО «ВСП», был совершен анализ текущего процесса по планирования и заказу компонентов, обоснована необходимость реализации информационной

системы по автоматизации процесса организации цепи поставок для ее улучшения.

Во второй главе разработаны модели информационной системы, обоснован выбор системы и её архитектуры, эффективности.

В заключении были подведены итоги по исследованию среды реализации информационной системы, сделаны выводы по проделанной работе.

В приложении приведены скриншоты приложения системы и части программного кода.

Глава 1 Анализ процесса заказа авто компонентов в ООО «ВСП»

1.1 Характеристика производственного предприятия

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ВАЛЕО СЕРВИС РОССИЯ» уже 11 лет работает на рынке в России.

Большой уставный капитал компании составляет 1.2 миллиарда рублей.

Генеральный директор - Пети Жан-Клод Этьен Виктор Мари.

Сайт компании: <https://www.valeo.com/en/>

Основным видом деятельности компании является торговля автомобильными деталями, узлами и принадлежностями, а именно стартеры, сцепления, системы кондиционирования, освещения, охлаждения и др.

ООО «ВСП» это международная компания, имеющая большой штат работников, с хорошим обеспечением материальными и техническими ресурсами, есть необходимое компьютерное оснащение, офисные помещения, транспорт, обеспечена орг. структура и управленческий персонал, специалисты, являющиеся профессионалами в своей деятельности.

Действует компания в г. Тольятти и имеет два филиала в России. В мире Валео имеет филиалы в 20 странах мира (143 завода).

Первый завод был построен в 1996 году в Польше, г. Скавина. В России первый завод был открыт в 2011 году в Нижегородской области. В 2012 было открыто производство в Тольятти и насчитывает 2 филиала на данный момент.

На данный момент предприятие в Тольятти осуществляет производство отопителей, радиаторов, конденсоров и генераторов для автомобилей.

По объему выручки Валео занимает 44 место в Самарской области и 2 место в отрасли.

По стоимости бизнеса занимает 51 позицию в регионе и 2 место в отрасли после Роберт Бош Самара.

Адреса филиалов в Тольятти:

Самарская обл., г. Тольятти, ул. Борковская, д.17

Самарская обл., г. Тольятти, ул. Северная, д.10а

Компания уже 11 лет на рынке, имеет хорошие отзывы от потребителей и стабильное финансовое положение.

На рисунке 1 представлена организационная структура ООО «ВСР»

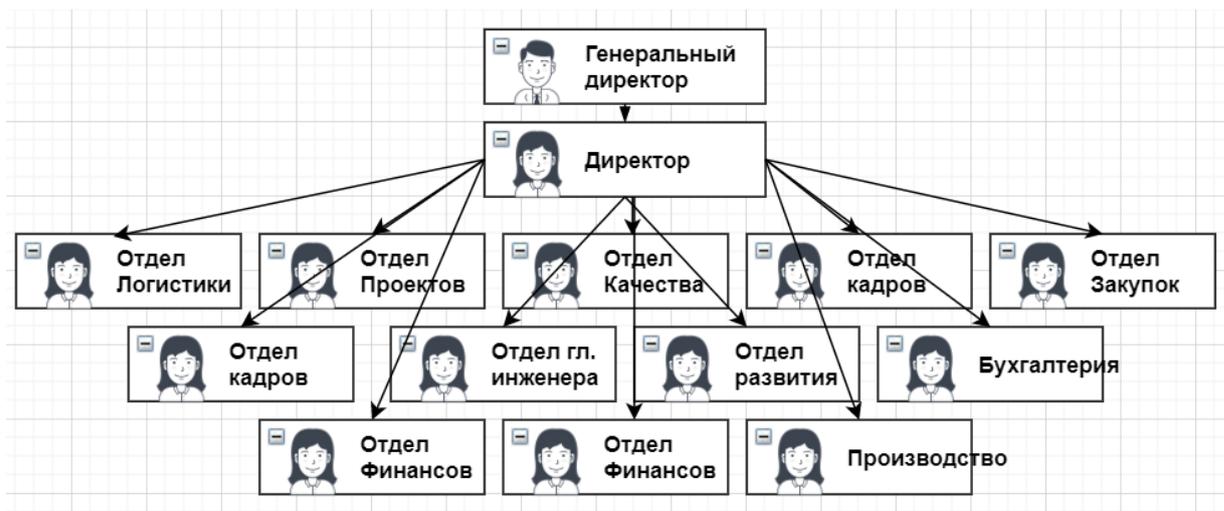


Рисунок 1 – Организационная структура ООО «ВСР»

В компании поставками и оптимизацией цепей поставок занимается отдел Логистики.

1.1.1 Характеристика процесса заказа производственных компонентов

Предметной областью выпускной квалификационной работы является процесс по заказу производственных компонентов.

На производственном предприятии ООО «ВСР» заказ компонентов происходит через систему SAP GUI.

SAPGUI – это клиент GUI в 3х –уровневой архитектуре SAP ERP базы данных, сервера приложений и клиента.

Это программное обеспечение, позволяющее пользователю получить доступ к функциональности SAP.

Пользователи должны аутентифицировать себя при доступе к SAP [17, с. 145]. Однако по умолчанию SAP использует незашифрованную связь, которая позволяет потенциальным злоумышленникам внутри компании получить доступ к именам пользователей и паролям, прослушивая сеть. Это может раскрыть всю систему SAP, если человек может получить доступ к этой информации для пользователя с расширенными полномочиями в системе SAP.

Это программное обеспечение позволяет учитывать количество компонентов на складе, в транзите и в сборке. Используя всю получаемую информацию система, сама определяет потребности в заказе компонентов для предприятия при правильных настройках [7, с.3].

Для этого процесс должен быть автоматизирован:

- все материалы должны быть заведены в системе;
- все продукты должны быть заведены в системе;
- все компоненты должны быть заведены в системе;
- все компоненты должны быть внесены в контракт и иметь quality module;
- иметь правильные настройки в зависимости от типа компонента;
- состоять в BOM (Bill of Material);
- потребности (Заказы) на продукт должны высылаться потребителем раз в неделю;
- прогнозы на продукт должны высылаться потребителем раз в неделю;
- необходимо обеспечить правильную работу e-Kanban.

Если все пункты соблюдены система может автоматизировать процесс заказа компонентов, но не транспортных компаний.

В настоящее время на предприятии не был внедрен ни один программный продукт для автоматизации поставок.

Для сокращения расходов на транспортировку на заводе используется отдельная должность – Аналитик цепей поставок. Для его оптимальной

работы необходима автоматизация процесса, так как для анализа поставок тратится большое количество времени и трудозатрат.

В большинстве случаев выбор перевозчика определяется несколькими параметрами:

- стоимость перевозки;
- время транзита;
- стоимость таможенного оформления;
- закрепенная репутация перевозчика;
- условия поставки.

Для того чтобы анализировать и выбрать наиболее выгодный для предприятия путь необходимо:

- письменно запросить всех перевозчиков, с которыми заключен контракт;

Данный процесс не занимает много времени, если рассчитывать, что перевозка будет одна и все данные нужно систематизировать для 1 груза. Но зачастую в одном грузе сочетается 10-30 позиций. И за всю практику предприятия в среднем 5-6 перевозок происходит одновременно только для серийно жизни, не считая проектную.

- подготовить бизнес переписку;

Бизнес переписка является одним из основных документов, влияющих на стоимость перевозки. Она содержит в себе данные об отправителе и получателе, весь список компонентов перевозки их стоимости и цене.

- подготовить Техническое описание продукта;

На предприятии ООО «ВСП» техническое описание составляется работником отдела RnD. Обычно все технические описания, для продукта, который уже перевозился ранее, хранятся в онлайн доступе на сайте. Но, в случае проектных поставок чаще всего большинство изделий является уникальными и не имеют аналогов. Поэтому для каждого референса сотрудник составляет свой документ, содержащий информацию о торговой

марке, производителе, весе, использованных материалах, фотографию изделия.

- составить инвойс;

Инвойс также является одним из обязательных документов, предоставляемых в таможенную для груза. В инвойсе обязательно указывать дату, уникальный номер инвойса, стоимость каждой позиции в евро и её название каждой позиции на русском и английском языках, общая стоимость изделий.

- определить упаковочный лист;

Упаковочный лист — это обязательный документ для таможенного оформления. В нем должны быть указаны максимально точные данные. Так как на основе упаковочного листа составляется таможенная декларация и в случае если есть расхождение по данным груз может быть остановлен и назначен дополнительный осмотр.

Дополнительный осмотр несет дополнительные расходы средств и времени и в большинстве случаев недопустим для предприятия.

В упаковочном листе обязательно указывать следующую информацию: вес нетто, общий вес нетто, габариты груза, вес брутто.

- связаться с отделом бухгалтерии и выслать документы по закупке товара;

Документы по закупке товара подтверждают стоимости компонентов, указанных в инвойсе и бизнес переписке и не являются обязательными, но в некоторых случаях таможенный представитель может запросить данный документ.

- определить коды ТНВЭД.

При составлении декларации коды ТНВЭД влияют на стоимость таможенных пошлин на ввоз/вывоз груза. Правильное определение кодов влияет на успешное и быстрое прохождение таможенной очистки.

Все заказы доставки компонентов осуществляются в «ручную» на основании собственного опыта. Чаще всего используется транспортная компания, которая уже ранее предоставляла свои услуги по перевозке. В связи

с этим, если цена уже ранее была согласована, нет необходимости её пересматривать.

Как следствие, не оптимальные цены для перевозки компонентов, так как грузы не консолидируются между поставщиками, находящимися недалеко друг от друга. Зачастую намного выгоднее объединять перевозки от поставщиков, находящихся в одной стране.

Для снижения затрат используется аренда автомобиля, который собирает компоненты от поставщиков из одной страны. Такой способ имеет название консолидация. Консолидация может также происходить между поставщиками из разных стран, объединяя маленькие поставки в одном пункте сбора, и оттуда в пункт назначения.

1.2 Концептуальное моделирование информационной системы по автоматизации процесса по организации цепей поставок производственного предприятия

1.2.1 Выбор методологии и технологии концептуального моделирования информационной системы

Для выбора методологии по работе и разработки концептуальной модели мы будем использовать методологию реинжиниринга.

Понятие реинжиниринга используется как пересмотр процессов и их переосмысление для улучшения показателей уже существующего бизнес процесса [1, с. 98]. Реинжиниринг сейчас является одной из самых действенных методологий для определения слабых мест процесса, их анализа и последующего улучшения.

В процессе данной методологии обычно выделяют 3 этапа, а именно:

- составление модели «КАК ЕСТЬ»;
- анализ модели «КАК ЕСТЬ»;

На этом этапе важно проанализировать все текущие процессы и найти слабые, уязвимые места процесса. Именно поэтому на предыдущем этапе

важно было не идеализировать входящие данные, так как последующий анализ некорректных данных был бы бесполезен.

– составление модели «КАК ДОЛЖНО БЫТЬ».

Моделью «КАК ДОЛЖНО БЫТЬ» называют модель состояния предприятия после преобразований, то есть после внедрения процессов по автоматизации [4, с. 54].

1.2.2 Моделирование бизнес-процесса «КАК ЕСТЬ» текущего процесса по заказу компонентов.

1.2.2.1 Разработка модели бизнес-процесса «КАК ЕСТЬ»

Моделью «КАК ЕСТЬ» называют модель текущего состояния предприятия [10, с. 37].

Разработка модели «КАК ЕСТЬ» может систематизировать данные и протекающие процессы. На основе этой модели могут быть выяснены уязвимые места в текущей модели. На этапе моделирования максимально важно приблизить данные к реально существующим, основываясь при этом на реальных процессах и ни в коем случае ни идеализировать существующие данные [3, с. 67].

В данном случае модель «КАК ЕСТЬ» будет обозначать неавтоматизированный процесс по заказу авто компонентов. В модель необходимо внести всю информацию, которая используется в процессе заказа компонентов от начала до конца.

Вся использованная информация будет являться исходными данными для последующего анализа.

Для составления модели были использованы следующие методы: сбор и анализ документации по логистике ООО «ВСП», опрос работников логистики (плановики производства, бригадиры, координаторы по проектной и серийной логистике) опрос поставщиков, интервьюирование перевозчиков.

На рисунке 2 представлена модель бизнес процесса по управлению цепями поставок по методологии «КАК ЕСТЬ».

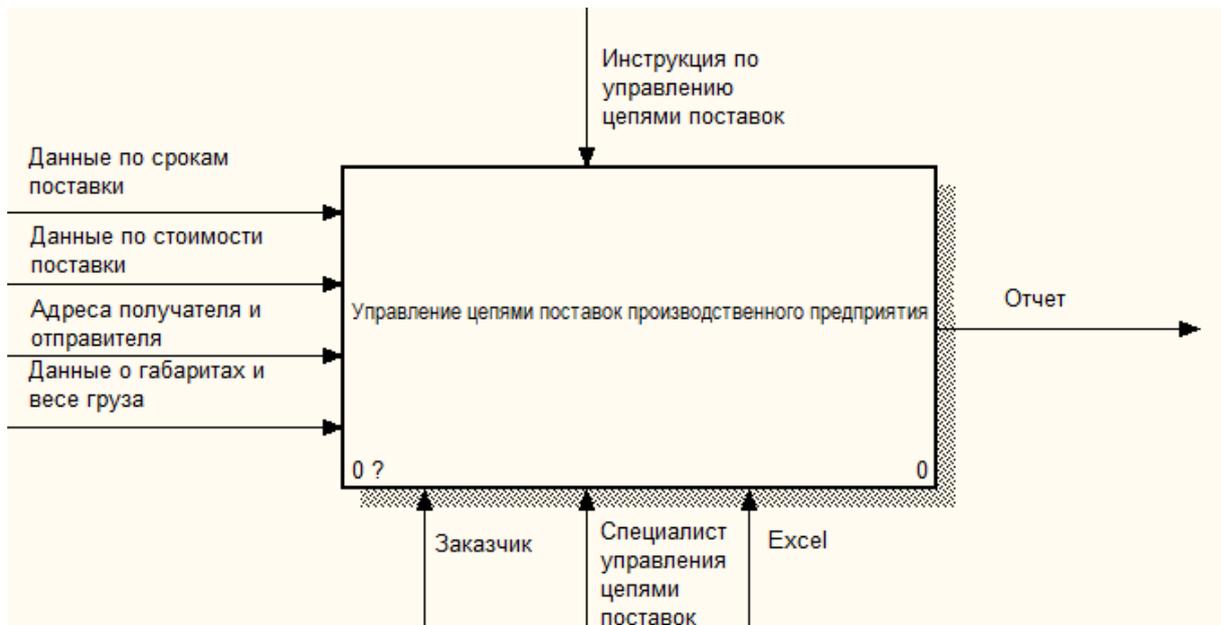


Рисунок 2 – Диаграмма «КАК ЕСТЬ» бизнес процесса по управлению цепями поставок (0-й уровень)

Входные данные бизнес-процесса:

- данные по срокам поставки;

Данные по срокам поставки в основном поступаю от менеджера по проекту исходя из Tooling Master list проекта.

Tooling Master List – документ содержащий в себе планирование всех процессов по доставке проекта от его начала до конца. Он учитывает доставку необходимого оборудования, готовых компонентов и их незавалидированных прототипов, материала, пресс форм, контрольно-измерительных приспособлений и так далее, и содержит информацию о вехах проекта к которым все вышеуказанное должно быть доставлено.

- данные по стоимости доставки;

Данные по стоимости в основном поступают от менеджера проекта совместно и менеджером по логистике, и в случае если речь идет о сумме, превышающей тысячу евро в обсуждение в обязательном порядке включается директор предприятия.

- адреса получателя и отправителя;

Адреса получателя и отправителя в основном поступают от заказчика отправки либо они уже есть в базе координатора по отправке.

– данные о габаритах и весе груза.

Данные о габаритах и весе груза поступают либо от бригадиров, упаковывающих груз, если груз отправляется с предприятия, либо от любого другого отправителя груза в зависимости от того где груз находится.

Механизмами бизнес-процесса управления цепями поставок являются: Заказчик, Специалист по управлению цепями поставок (СУЦП) и Табличный процессор (ТП) Excel.

На выходе формируются данные о наиболее выгодной цепи поставок.

Управление осуществляется в соответствии с Инструкцией по управлению цепями поставок.

Декомпозиция бизнес-процесса управления цепями поставок представлена на рисунке 3.

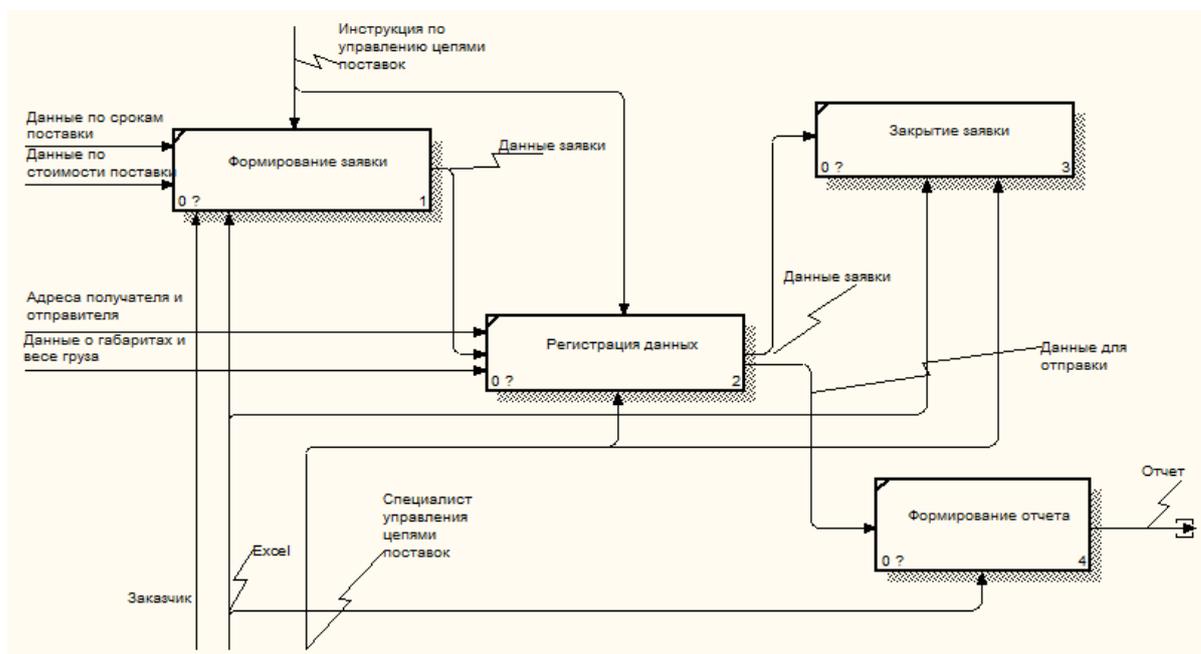


Рисунок 3 - Декомпозиция бизнес-процесса управления цепями поставок «КАК ЕСТЬ» (1-й уровень)

Анализ декомпозиции бизнес-процесса управления цепями поставок показал, что существующий процесс не является оптимальным для решения поставленных задач.

1.2.2.2 Разработка модели бизнес-процесса «КАК ЕСТЬ»

Анализ модели «КАК ЕСТЬ» управления цепями поставок предприятия «КАК ЕСТЬ» показывает, что в процессе существуют свои недостатки, а именно:

- журнал заявок разбит на два дополнительных журнала по одобрению: для одобрения стоимости отправки и для одобрения сроков доставки;
- журнал на базе табличного процессора Excel не обладает достаточным функционалом для ведения корректного учета;
- отсутствие нормализованной реляционной базы данных ухудшает работу Журнала заявок из-за двойной траты времени на одобрение;
- табличный процессор Excel не имеет возможности доступа нескольких человек одновременно;
- табличный процессор Excel не имеет возможности удаленного доступа к журналам одобрения.

Учитывая вышеизложенные недостатки предприятие приняло решение об улучшении текущего процесса по управлению цепями поставок используя новую информационную систему управления цепями поставок.

1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

- После проведенного анализа существующего процесса по заказу авто компонентов были выявлены следующие недостатки:
- журнал заявок разбит на два дополнительных журнала;
- использование ТП Excel не обладает необходимым функционалом;
- отсутствует нормализованный процесс по одобрению входных данных.

Согласно проведенному анализу делаем вывод о том, что необходимость разработки автоматизированного варианта решения является целесообразной, поскольку он позволит устранить недостатки текущего процесса.

После выявления недостатков существующего бизнес-процесса, необходимо описать требования к проектируемой информационной системе.

Основываясь на требованиях заказчика и анализе процесса, в таблице 1 были сформированы требования к проектируемой информационной системе.

Таблица 1 – Требования к информационной системе

Требование	Статус	Полезность
Интуитивно понятный интерфейс	Одобрено	Высокая
Возможность добавление новых поставщиков	Предложенные	Средняя
Возможность удаленного доступа	Одобрено	Высокая
Возможность доступа нескольких человек одновременно	Одобрено	Высокая
Круглосуточная и бесперебойная работа	Одобрено	Высокая
Возможность резервного копирования данных	Предложенные	Средняя
Время отклика системы – не более чем 5 секунд	Одобрено	Высокая
Адаптивный дизайн	Предложенные	Низкая
Возможность обеспечения одновременной работы не менее 300 пользователей.	Одобрено	Высокая
Резервное копирование раз в месяц	Предложенные	Средняя
Сервер приложений и СУБД должны функционировать на одном или нескольких выделенных серверах	Одобрено	Высокая
Клиентская часть системы должна работать только на специально отведенных предприятием компьютерах сотрудников и администраторов системы	Одобрено	Высокая
Возможность выбора получателя и отправителя из списка	Одобрено	Высокая
Создание отчета автоматически	Одобрено	Высокая

После описания требований и определения их эффективности, необходимо проанализировать преимущества и недостатки существующих технологий в данной предметной области.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

1.3.1 Определение критериев анализа

Для определения критериев анализа необходимо провести сравнительный анализ наиболее известных информационных систем для управления цепями поставок. Это необходимо для того, чтобы оценить плюсы и минусы существующих решений и учитывать это при выборе программного продукта для разработки.

Далее рассмотрим существующие аналоги информационных систем для управления цепями поставок.

Первым для рассмотрения будет программный продукт iScala. Обновленный интерфейс iScala 3.1 изображен на рисунке 4.



Рисунок 4 – Пользовательский интерфейс iScala 3.1

iScala это продукт компании Epicor, предназначенный для управления различными деятельностью предприятия, такими как финансы, материальное снабжение, сервисное обслуживание, управление производством и так далее.

Основными преимуществами параметрически настраиваемой системы iScala являются:

- универсальность,
- качественная поддержка,
- частое обновление версий.

К минусам такой системы относятся:

- недостаточная гибкость,
- быстроедействие ниже чем у других систем.

Далее перейдем к рассмотрению программного продукта Netsuite. Пользовательский интерфейс NetSuite изображен на рисунке 5.

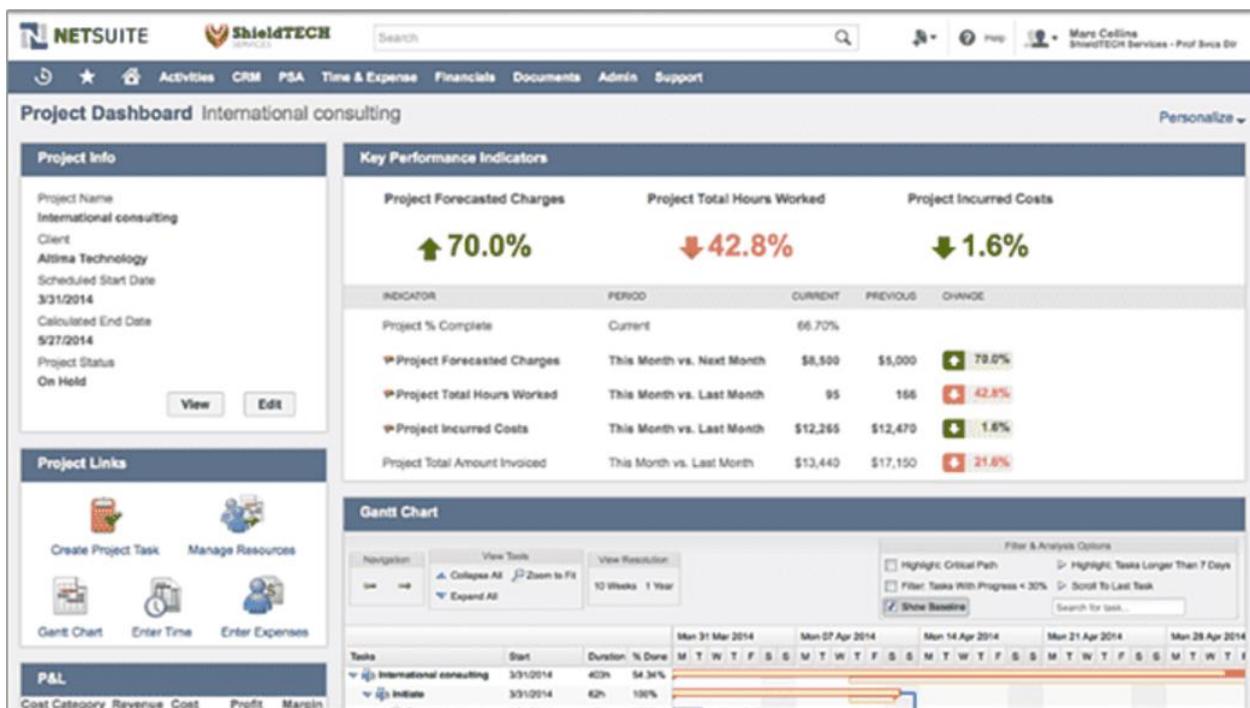


Рисунок 5 – Пользовательский интерфейс NetSuite

NetSuite это новая информационная система от компании Systematic Software Solution, создание которой поддержал Ларри Эллисон, основатель Oracle.

Основными данной информационной системы являются:

- хорошая интегрируемость;
 - работа не требует наличие специально выделенного сервера;
 - бесплатная лицензия;
 - возможность делать небольшие настройки самостоятельно.
- К минусам такой системы относятся:
- у пользователей нет прав на ПО;
 - необходимо стабильное соединение с Интернет;
 - оплата аренды приложений в зависимости от количества пользователей.

Последним для рассмотрения будет являться программный продукт SAP. Пользовательский интерфейс SAP изображен на рисунке 6.

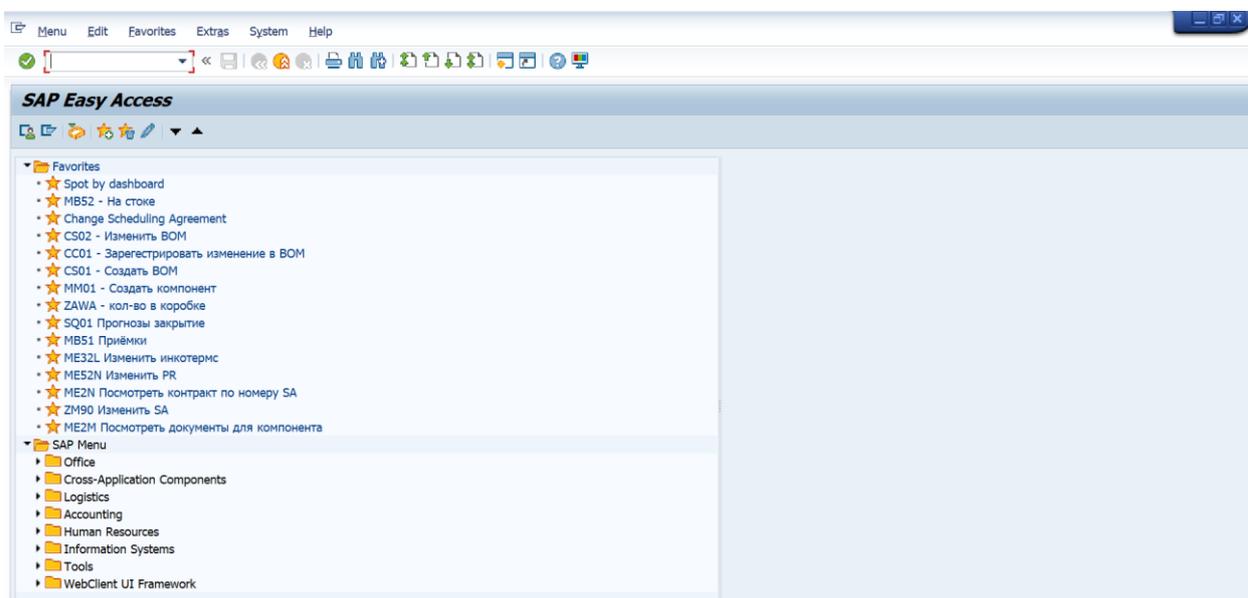


Рисунок 6 – Пользовательский интерфейс SAP

Система SAP является программным решением для автоматизации и стандартизации процессов бухгалтерии, торговли, казначейства, логистики.

Плюсами SAP можно назвать:

- обработка больших объемов данных;
- подходит для международного использования;
- стандартизированный процесс.

Минусами являются:

- большие затраты на внедрение;
- не интуитивно понятный интерфейс;
- для работы необходимо специальное обучение;
- отсутствие гибкости.

Система управления цепями поставок SAP используется на предприятии ООО «ВСП»

1.3.2 Сравнительная характеристика существующих разработок

Анализ существующих информационных систем показан в таблице 2.

Таблица 2 – Сравнение существующих разработок

Критерий	iScala	NetSuite	SAP
Универсальность	+	-	+
Наличие качественной поддержки	+	-	+
Гибкость	-	+	-
Быстродействие	-	+	+
Низкая стоимость внедрения	+	+	-
Интуитивно понятный интерфейс	+	+	-
Обработка больших объемов данных	-	-	+
Простое обучение для работы пользователя	+	+	-
Международное использование	-	-	+

Исходя из сравнения существующих ERP- систем сделать вывод, что у каждой из представленных систем есть свои преимущества и недостатки для управления цепями поставок.

1.4 Постановка задачи на разработку проекта создания/внедрения информационной системы автоматизации процесса по организации цепей поставок

Исходя из анализа, проведённого в пункте 1.3.7 можно сделать вывод, что ни одна из рассмотренных ERP систем в данной предметной области не удовлетворяет всем необходимым требованиям. Следовательно, еще раз подтверждается необходимость проектирования и разработки новой информационной системы для управления цепями поставок.

Целью создания информационной системы является:

- создание интуитивно понятного для конечного пользователя программного продукта;
- сокращение времени на одобрение;
- сокращение рутинной работы на переписке с перевозчиками;
- установление нормализованного процесса для управления поставками.

Для достижения поставленной цели были выведены следующие требования:

- интуитивно-понятный интерфейс;
- возможность удаленного доступа;
- возможность работы нескольких человек одновременно;
- автоматическое создание отчета.

Специалист по управлению цепями поставок на предприятии «ВСП» тратит много времени для того чтобы организовать наиболее выгодный для компании логистический поток. Именно поэтому необходимо автоматизировать процесс для сокращения рутинной работы и внедрить информационную систему управления цепями поставок, которая позволит сократить время затраты на определении выгодного потока.

1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

Диаграмма бизнес-процесса управления цепями поставок «КАК ДОЛЖНО БЫТЬ» изображена на рисунке 7.



Рисунок 7 - Контекстная диаграмма бизнес-процесса управления цепями поставок «КАК ДОЛЖНО БЫТЬ» (0-й уровень)

Изменения в диаграмме выделены красным цветом.

Декомпозиция указанного бизнес-процесса приведена на рисунке 8.

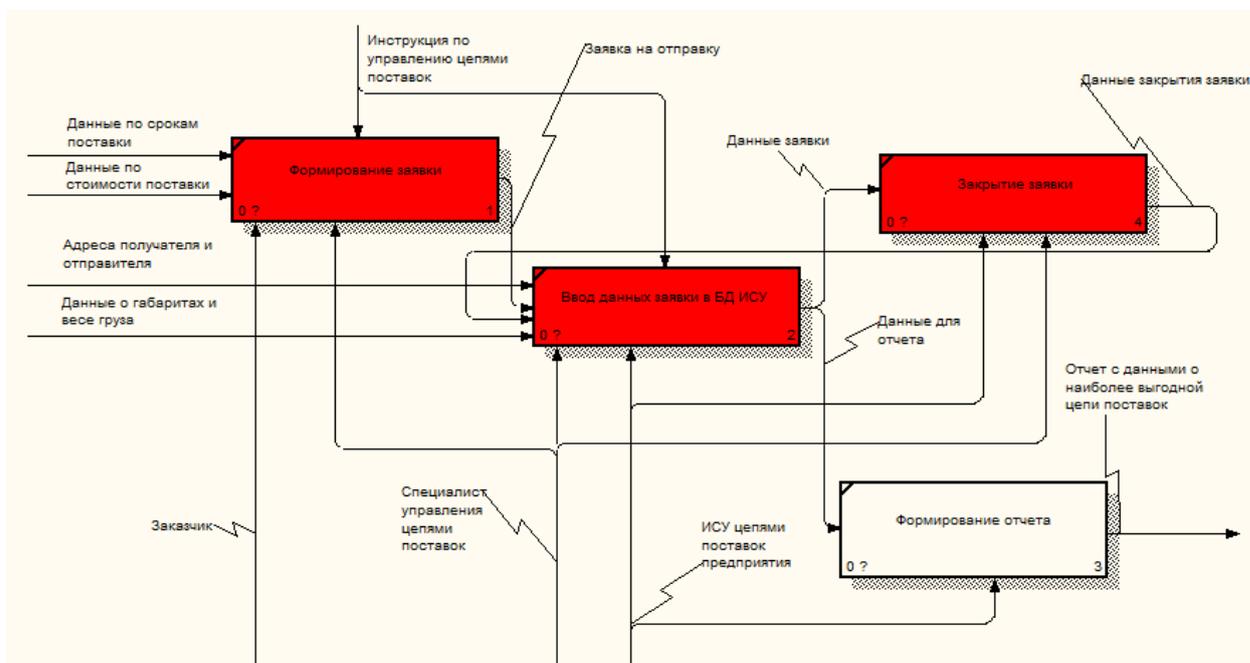


Рисунок 8 - Декомпозиция бизнес-процесса управления цепями поставок «КАК ДОЛЖНО БЫТЬ» (1-й уровень)

Опишем элементы, изображенные на диаграммах нулевого и первого уровня.

Входные данные:

- данные о сроках;
- данные о стоимости;
- данные о получателе и отправителе;
- данные о габаритах и весе груза.

Выходные данные:

- автоматический отчет заказчику с данными о наиболее выгодной цепи поставок.

Управляющие воздействия:

- инструкция по управлению цепями поставок предприятия.
- исполнители:
 - заказчик,
 - специалист управления цепями поставок,
 - информационная система управления цепями поставок.

Перейдем к описанию реинжиниринга процесса.

Заказчик заводит заявку на основе входных данных о габаритах и весе груза, получателя, отправителя, стоимости и сроках отправки;

Производится обработка заказа на предмет того, существует ли уже поток по данному отправлению. Если потока не существует, то формируется заявка на отправку. В противном случае отправка использует текущий поток.

Автоматический отчет с результатами заявки отправляется заказчику с указанием стоимости доставки и ближайшей датой поставки.

Заполненная заявка заносится в базу данных информационной системы управления цепями поставок и ей присваивается статус «в процессе».

После выполнения заявки формируется реальные данные выполнения заявки и ее статус переходит в «выполнено».

Выводы по главе 1

Первая глава представляет собой описание функциональной модели предметной области. Был сделан выбор технологии концептуального моделирования, а также были разработаны модели бизнес-процессов «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ». Модель «КАК ЕСТЬ» была использована для описания существующего процесса. После анализа указанных моделей был сделан вывод и необходимости автоматизации текущего процесса по заказу компонентов.

Также был проведен сравнительный анализ существующих информационных систем для управления цепями поставок и их сравнение. Анализ подтвердил, что ни одна из рассмотренных информационных систем не удовлетворяет всем необходимым требованиям.

Исходя из этого делаем вывод что необходима разработка новой информационной системы.

Были сформированы и описаны требования к проектируемой информационной системе, после этого была выполнена постановка задачи на разработку новой информационной системы.

Глава 2 Логическое проектирование информационной системы автоматизации процесса организации цепей поставок производственного предприятия

2.1 Выбор технологии логического моделирования информационной системы

Логическое моделирование решений информационной системы управления цепями поставок включает в себя:

- разработку объектной модели автоматизированной информационной системы;
- разработку логической модели автоматизированной информационной системы;
- разработку базы данных.

Разработка объектной и логической моделей будет представлена с использованием унифицированного языка моделирования (UML). Данная система обозначений в основном применяется для объектно-ориентированного проектирования. UML не только предоставляет возможность рассмотреть поставленную задачу с разных сторон, но и является простой для чтения других программистов [2, с. 121].

Разработка базы данных должна происходить на основе концептуальной модели, составленной ранее.

2.1.1 Разработка требований к информационной системе

Используя составленную ранее модель «КАК ДОЛЖНО БЫТЬ» и требования, которые были предъявлены заказчиком к информационной системе управления цепями поставок должна быть разработана диаграмма вариантов использования бизнес процесса цепями поставок предприятия.

Диаграмма вариантов использования включает в себя описание окружающей среды предприятия, включающую в себя заказчиков, перевозчиков, поставщиков и так далее, основные бизнес процессы, а также

взаимодействие между ними. Диаграмма вариантов использования представлена на рисунке 9.

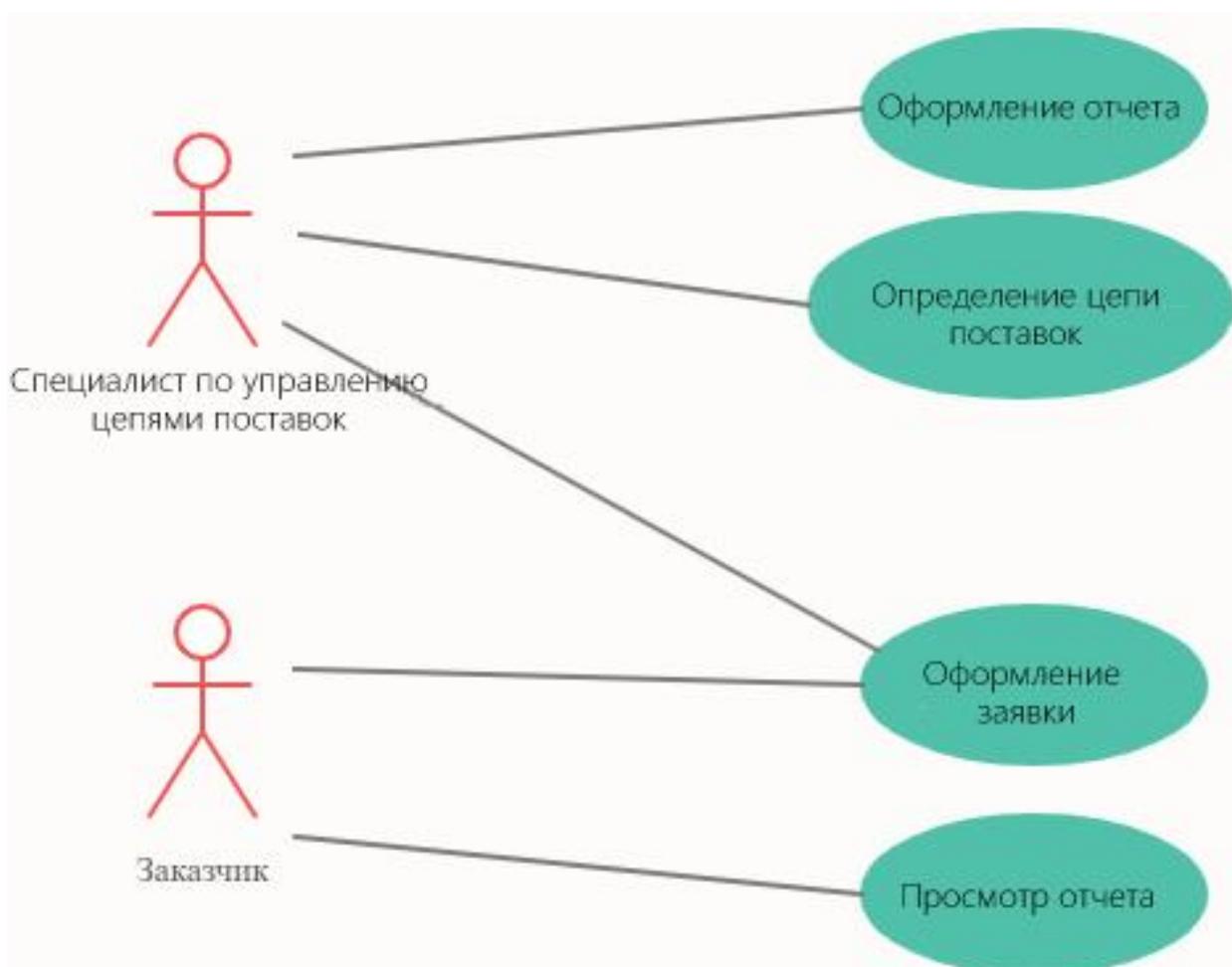


Рисунок 9 – Диаграмма вариантов использования

Основным действующим лицом бизнес процесса является заказчик, то есть человек, которому необходимо сформировать отчет о самых выгодных условиях поставки.

Можно выделить основные прецеденты, которые необходимо реализовать в системе. Описание прецедентов, использованных в системе изображено на таблице 3.

Таблица 3 – Описание прецедентов системы

Прецедент	Описание
Формирование заявки	Использование входных данных для составления заявки
Ввод данных заявки	Регистрация заявки в БД ИСУ
Закрытие заявки	Ввод всех вводных данных о согласовании поставки
Формирование отчета	Автоматизированное формирование и вывод отчета

Требования к информационной системе представляют из себя совокупность предположений относительно свойств, атрибутов, функционала и качеств программного продукта, который подлежит интегрированию в деятельность организации [5, с. 243].

2.1.2 Разработка диаграммы классов информационной системы управления цепями поставок производственного предприятия.

Для статического описания информационной системы с точки зрения ее разработки будет использована диаграмма классов. Диаграмма классов содержит в себе основные классы и интерфейсы и показывает взаимосвязь между ними [6, с. 347].

Основными достоинствами такой диаграммы является сокращение время затрат на объяснение задачи, наиболее наглядная визуализация структуры действующих элементов и их взаимодействие.

На рисунке 10 изображена диаграмма классов информационной системы управления цепями поставок производственного предприятия.

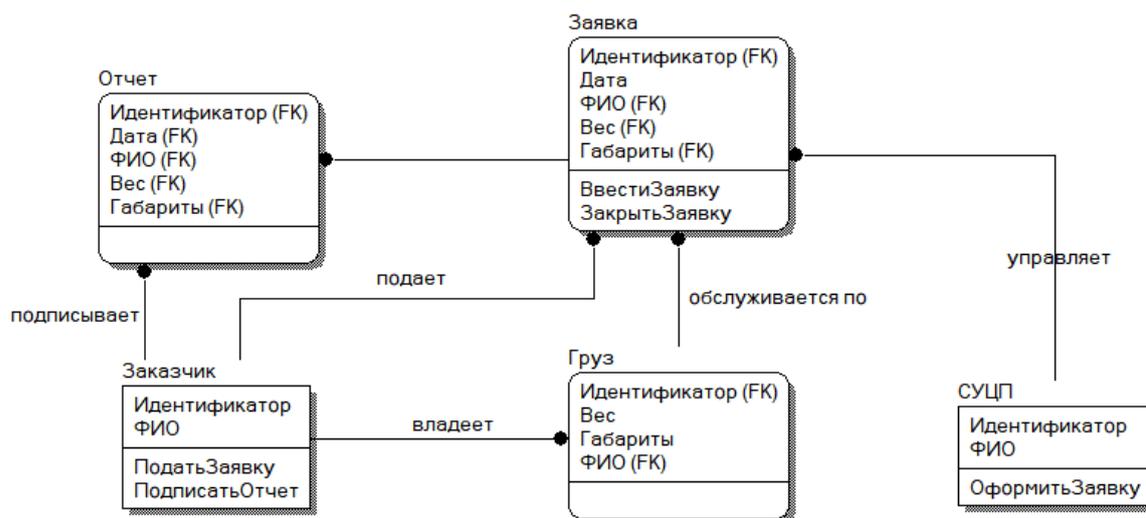


Рисунок 10 - Диаграмма классов информационной системы управления цепями поставок производственного предприятия

Описание классов:

- заказчик – класс лиц, подающих заявки на формирование цепи поставок;
- специалист управления цепями поставок – класс лиц, оформляющих заявки на формирование цепи поставок Заказчика;
- заявка- класс документов, содержащий информацию об отправителе, получателе, информацию о грузе и дате поставки;
- отчет – класс документов, содержащий информацию о наиболее выгодной цепи поставок для выбранных условий.

Составленная выше диаграмма помогает определить основные ветки для разработки программного обеспечения и логической модели информационной системы автоматизации процесса организации цепей поставок предприятия.

2.1.3 Разработка диаграммы последовательностей управления цепями поставок производственного предприятия

Для определения основных участников процесса и последовательности их действий для управления, стадий от начала заявки и до конца будет использовано построение диаграммы последовательностей.

Основные элементы диаграммы последовательностей:

- объекты,
- линии жизни,
- стрелки.

На рисунке 11 показана диаграмма последовательностей оформления заявок на создание цепи поставок предприятия.

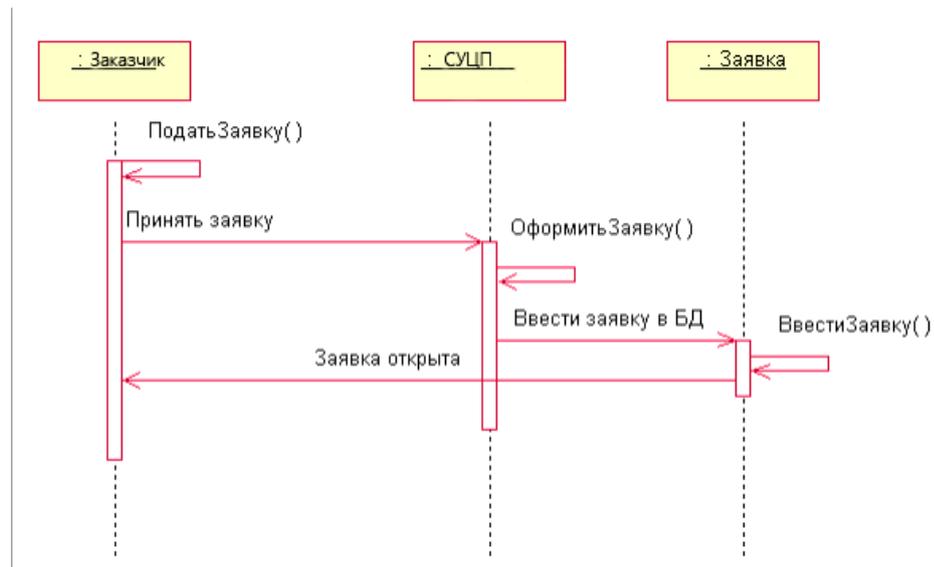


Рисунок 11 – Диаграмма последовательностей оформления заявок на создание цепи поставок предприятия

Последовательность действий показывает:

- обращение объекта Заказчик к объекту Специалист по управлению цепями поставок для приёма заявки;
- создание объектом Специалист по управлению цепями поставок заявки;
- регистрация заявки объектом Заявка;
- отправка сообщения от Специалиста по управлению цепями поставок Заказчику содержащем информацию об открытии заявки;
- завершение процесса оформления Заявки.

На рисунке 12 изображена диаграмма последовательностей завершения заявки на формирование цепи поставок.

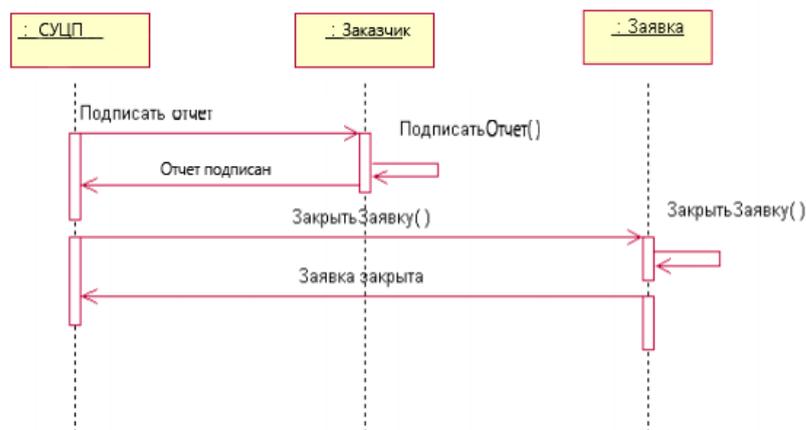


Рисунок 12 – Диаграмма последовательностей завершения заявки на формирование цепи поставок

Описание последовательностей действия:

- обращение объекта Специалист по управлению цепями поставок к объекту Заказчик для одобрения входных данных;
- обращение объекта Заказчик к объекту Специалист по управлению цепями поставок с подтверждением данных;
- регистрация закрытия проходит объектом Заявка;
- обращение объекта Заявка к объекту Специалист по управлению цепями поставок с информацией о регистрации закрытия заявки;
- завершение процесса закрытия Заявки.

Разработанные диаграммы последовательностей и диаграммы классов будут использованы для построения логической модели данных автоматизированной информационной системы.

2.2 Информационное обеспечение информационной системы

2.2.1 Используемые классификаторы и системы кодирования

В рамках информационного обеспечения автоматизированной информационной системы необходимо рассмотреть классификаторы экономической информации.

Классификаторы и их системы их кодирования используются в основном для заполнения документов, подготовки отчетов.

Классификаторы могут быть разработаны как для определенной компании, так и для государства. Уровни классификаторов различают:

- международные,

Классификаторы, использующиеся во всех странах

- межгосударственные,

Классификаторы, использующиеся в межгосударственных объединениях, такие как ЕС, СНГ.

- национальные,

Классификаторы, которые не только должны использоваться только в пределах государства, но и должны не противоречить международным.

- отраслевые,
- Классификаторы, использующиеся в одной отрасли.
- системные.

Классификаторы, используемые конкретной кампанией для применения в автоматизированной информационной системы.

Для обеспечения более эффективной обработки данных, сокращение общего объема работы используется система кодирования. Кодом называется обозначение какого-либо объекта в виде знака. Сам процесс присвоения кода и есть кодирование [9, с. 321]. Используемые классификаторы и коды изображены на таблице 4.

В таблице были распределены прецеденты по типу значимости, системы кодирования, коду и значению кода. В процессе были выведены следующие коды: код поставщика, код перевозчика и код страны.

Таблица 4 – Используемые классификаторы и коды

Наименование объекта	Значимость кода	Система кодирования	Код	Значение
Код поставщика	5	П	01-55	Порядковые номера поставщиков
Код перевозчика	5	П	02-05	Порядковые номера перевозчиков
Код страны	10	П	03-08	Порядковые номера стран

Разработанные классификаторы позволяют упростить работу системы и повысить ее эффективность.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Входной информацией называется вся информация, которую получает автоматизированная информационная система для выполнения своих функций, то есть все вводимые данные со стороны электронно-вычислительной машины, человека, сигналов, либо из документов, различных носителей.

Входную информацию можно разделить по признаку источника на внутреннюю и внешнюю. Если информация была получена в процессе учета, регистрации и так далее, то она является внутренней. Если информация была получена извне, то есть от поставщика, потребителя, любой другой вышестоящей организации, она является внешней.

Внешняя информация часто может не быть четко определена, но внутренняя всегда должна реально отражать действительность.

Частью внутренней информации является инструкции, нормативные акты, законодательные документы. Именно такая информация поступает в

автоматизированную информационную систему для решения поставленной задачи.

На предприятии ООО «ВСП» все заявки регистрируются в журнале одобрения заявок. К входной информации относятся:

- фотографии компонентов,
- товарно-транспортная накладная,
- инвойс либо счет фактура,
- акт приема передачи,
- описание товара,
- бизнес переписка,
- прайс лист.

Все указанные в списке позиции являются обязательными для внесения в журнал.

2.2.3 Характеристика выходной информации

Выходной информацией называется информация, которая была получена в результате работы автоматизированной информационной системы. Выходная информация может быть представлена в виде:

- файла;
- базы данных;
- банка данных.

Файлы можно разделить на переменные и постоянные. Переменными называются те, что могут изменяться в течение года на двадцать процентов. Файлы, которые содержат всю остальную информацию являются постоянными.

Базы данных состоят из: базовых файлов, рабочих файлов, файлов связи.

Пользователи, использующие результаты выходной информации:

- специалист управления цепями поставок;
- заказчик.

Выходная информация автоматизированной информационной системы автоматизации процесса организации цепей поставок производственного предприятия представлена на рисунке 13.

Отчет	
Поручение Экспедитору/Перевозчику	
Дата поставки:	17.05.2020
Заказчик/Клиент:	ООО "BCP", РФ, 445000, Самарская область, г. Тольятти, ул. Борковская, 17.
Экспедитор/Перевозчик:	SCHENKER
Наименование грузоотправителя:	Valeo Togliatti
Адрес грузоотправителя:	Russia, Togliatti, Borkovskaya 17
Наименование грузополучателя:	Nidec Spain
Адрес грузополучателя:	Spain, STA PERPETUA DE MOGODA POLIGONO INDUSTRIAL CAN VINUYALS CALLE GUIFRE EL PILOS, S/N
Информация о грузе:	123
Путь:	Russia - Poland - Spain
Стоимость перевозки (включая экспортное оформление в стране отправления):	466

Рисунок 13 - Выходная информация информационной системы автоматизации процесса организации цепей поставок предприятия

Отчет, изображенный на рисунке, показывает данные о наиболее выгодном варианте перевозки груза. В нем обязательно должны быть указаны полное название и адрес отправителя и получателя, выбранный перевозчик, информация о грузе, и стоимость перевозки.

2.3 Проектирование базы данных информационной системы автоматизации процесса организации цепей поставок производственного предприятия.

2.3.1 Выбор технологии проектирования базы данных информационной системы автоматизации процесса организации цепей поставок производственного предприятия

Написание программы в основном начинается с построения концептуальной модели.

Концептуальная модель - это модель показывает множество сущностей и связей между ними, определяющих смысл рассматриваемой предметной области. Данная модель помогает определить важность объектов и связей между ними и является связующим элементом между теорией и наблюдениями.

Для построения концептуальной модели необходимо сформулировать цель и задачи, определить объект и предмет исследования. На данном этапе необходимо провести анализ научной литературы и структуру объекта и максимально точно интерпретировать основные понятия.

Концептуальная модель может быть представлена в графической форме и в форме описания, но предпочтение отдается графическому представлению с помощью методологии IDEFX1.

2.3.2 Разработка концептуальной модели данных информационной системы

Целью разработки концептуальной модели данных является представление информации в наиболее удобной для конечного пользователя форме. Поэтому для построения концептуальной модели используют язык, наиболее схожий с естественным.

На этапе концептуального моделирования автоматизированной информационной системы выделяют три этапа:

- описание общего представления предметной области;
- описание функций и задач автоматизированной информационной системы;
- описание основных сущностей и связей.

Концептуальная модель данных автоматизированной информационной системы представлена на рисунке 14.

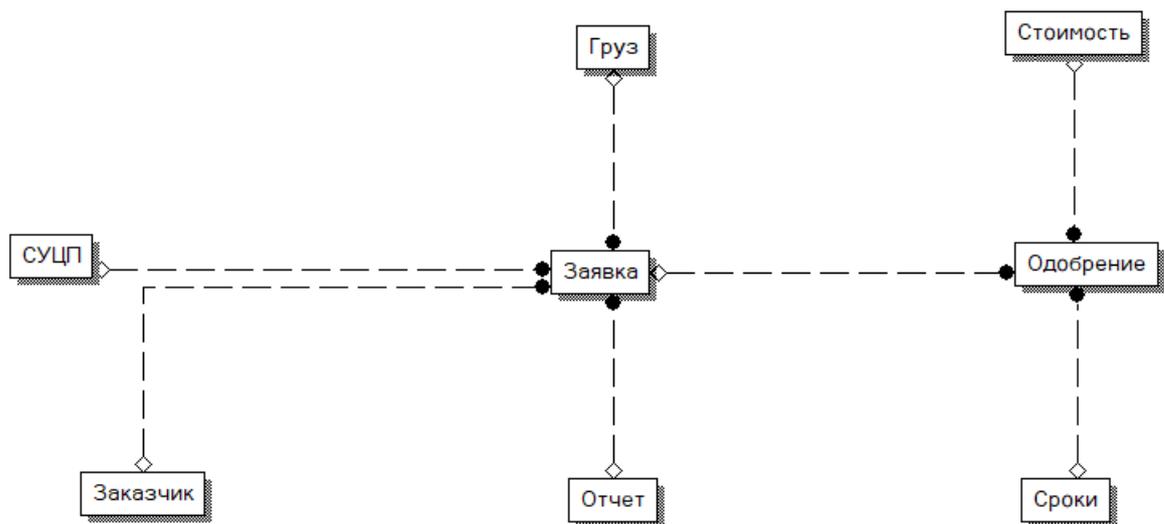


Рисунок 14 - Концептуальная модель данных информационной системы автоматизации процесса организации цепей поставок

Основными объектами модели являются сущности и связи между ними. Объектами являются:

- заказчик;
- специалист управления цепями поставок;
- отчет;
- информация и сроке поставки;
- информация и грузе;
- информация об отправителе;
- информация о получателе;
- информация о стоимости поставки.

Между сущностями модели в рамках одного бизнес-процесса установлены следующие связи: Заказчик может иметь несколько Заявок («один ко многим»); Специалист управления цепями поставок может оформить несколько Заявок («один ко многим»); по Заявке формируется только Отчет («один к одному»); по Заявке обслуживается только один груз («один к одному»). Все связи на диаграмме не идентифицирующие.

2.3.3 Обоснование вида логической модели

Информационная система, рассматриваемая в рамках автоматизации процесса организации цепей поставок основана на реляционной системе управления базами данных, и выполняется средствами ER модели данных «сущность-связь».

Модель «сущность-связь» является одним из средств моделирования предметной области на данном этапе. Название ER происходит от английских слов: Entity – сущность и Relation – связь [11, с. 88].

ER модель сущность связь необходима для наиболее точного отображения реальной информации и используется для построения логической модели данных.

Основными объектами модели являются:

- сущность,
- атрибут,
- связь.

Сущностью называют некоторый объект реального мира [12, с. 277]. Данный объект может существовать независимо. Сущности изображаются прямоугольником с названием сущности.

Атрибутом называют свойства сущности.

Связь изображается ромбом с названием связи.

Часто для изображения модели используются CASE средства со стандартом IDEF1X, такие как:

- Erwin,
- design IDEF,

- power Designer.

Построение модели будет выполнено с помощью методологии IDEF1X.

Для нормализации связей между будущими объектами базы данных необходимо построить ER модель.

2.3.4 Разработка логической модели данных информационной системы

Логическая модель является наиболее показательной для представления логической структуры базы данных. Она не обязательная должна на сто процентов соответствовать готовой системе управления базами данных.

Основными объектами логической модели являются:

- сущности баз данных;

Сущности не являются объектами баз данных, они лишь дают описание этих объектов/субъектов. Для описания свойств объектов используются атрибуты [8, с. 52].

- связи между сущностями баз данных.

Для того чтобы показать взаимодействие объектов между собой используются связи. При наличии каких-либо ограничений во взаимодействии используются свойства связи.

Для оформления логической модели базы данных будет использовано преобразование UML диаграммы с использованием методологии IDEF1X на основании диаграмм последовательностей и классов, разработанных ранее.

На рисунке 14 показано логическая модель информационной системы управления цепями поставок предприятия.

2.4 Требования к аппаратно-программному обеспечению информационной системы управления цепями поставок производственного предприятия

После проведения анализа существующей модели предприятия и требований Заказчика можно вывести параметры, которыми должна обладать информационная система управления цепями поставок:

- учитывая современные технологические платформы и требования реализация программного продукта будет проходить на базе “MySQL Community Server 8.0.20”;
- главным требованием любой компании является внедрение улучшений за минимальную стоимость, поэтому используя данный продукт можно легко снизить затраты на внедрение и последующую эксплуатацию и обучение сотрудников;
- для успешной эксплуатации программного продукта необходим интерфейс имеющий максимально простой для обычного пользователя дизайн;
- в программном продукте должен быть реализован удаленный доступ для максимальной эффективности.

Учитывая вышеизложенные параметры системы среди аналогов было выбрано внедрить информационную систему управления (далее ИСУ) цепями поставок.

Выводы по главе 2

На первом этапе была разработана логическая модель данных на основе методологии IDEFX1. Для построения логической модели были разработаны диаграммы классов и последовательностей.

Далее было охарактеризовано информационное обеспечение автоматизированной информационной системы, были описаны

классификаторы и системы их кодирования, было выполнено описание входной и выходной информации базы данных.

На следующем этапе был сделан выбор технологии проектирования и была разработана концептуальная модель данных автоматизированной информационной системы. Описание концептуальной модели было реализовано с помощью CASE-средств.

Для реализации проекта были выбраны реляционной системы управления базами данных MySQL 8.0.18 и PHP 7.2.25.

На последнем этапе были описаны требования к аппаратно-программному обеспечению автоматизированной информационной системы.

Глава 3. Физическое проектирование информационной системы управления цепями поставок

3.1 Выбор архитектуры информационной системы управления цепями поставок

Согласно требованиям Заказчика, проектируемая информационная система управления цепями поставок должна обеспечивать одновременную работу не менее 300 пользователей, то есть подразумевает многопользовательную работу.

В связи с этим приложение необходимо реализовать на архитектуре «клиент-сервер». Модель взаимодействия данной архитектуры представлена на рисунке 16.

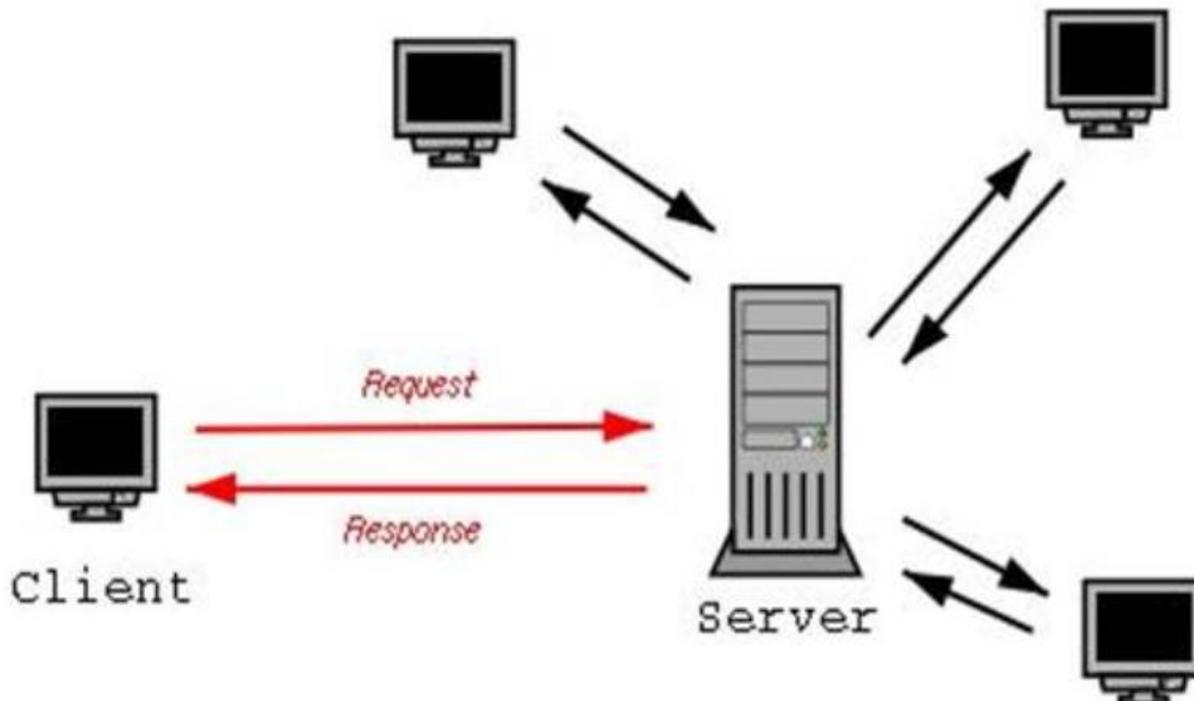


Рисунок 16 - Архитектура «клиент-сервер»

Серверами могут выступать все http сервера, такие как:

- MySQL,
- AMPPS,
- Apache.

Взаимодействие между Клиентом и Сервером происходит с помощью сетевых протоколов.

Сетевой протокол, или прокол передачи данных это набор правил для соединения и обмена данных между компьютерами [13, с.146]. В модели OSI сетевые протоколы делятся на 7 уровней:

- прикладной,
- представления,
- сеансовый,
- транспортный,
- сетевой,
- канальный,
- физический.

Одним из самых распространённых протоколов является TCP/IP.

TCP/IP расшифровывается как «Transmission Control Protocol/Internet Protocol» и в переводе означает Протокол управления передачей/Протокол Интернета [14, с. 568].

Стек данных протоколов был создан в 1972 году, и с тех пор обрел свою популярность среди пользователей из-за того, что большинство операционных систем и рабочих станций могут поддерживать протокол TCP/IP.

Изначально данный проект назывался Apranet, но после своего развития и изменения первоначальных задач в 1997 году он получил название Интернет. Руководитель проекта Винтон Серф за свой вклад получил Национальную медаль за заслуги в области технологий, также, как и его коллега Роберт Кан.

3.2 Выбор технологии разработки программного обеспечения информационной системы управления цепями поставок

Под термином технология разработки программного обеспечения обычно понимают список мер необходимых для разработки какого-либо программного продукта [20, с. 39].

Разработка любого программного обеспечения включает в себя много этапов. Для описания этапов необходимо ввести основное понятие в технологии разработки – жизненный цикл программного продукта.

Жизненный цикл - это период времени от момента, когда было принято решение о создании программного продукта, был проведен анализ требований Заказчика до того времени, как продукт перейдет в эксплуатацию конечным пользователем [18, с. 47].

В жизненном цикле любой программы можно выделить следующие процессы:

- приобретения,
- поставки,
- разработки,
- эксплуатации,
- сопровождения.

На первом этапе необходимо чтобы Заказчик подробно описал требования к конечному продукту, так как именно от них будет строиться весь процесс разработки. Также должны быть описаны и ограничения, такие как ограничение по количеству пользователей, архитектуре и условиям работы. После описания всех необходимых требований и ограничений заключается договор на разработку программного продукта.

На следующем этапе обычно проводится анализ требований и ограничений Заказчика, выбирается архитектура, удовлетворяющая требованиям, выбирается метод тестирования и контроля качества.

После непосредственной разработки программного продукта процесс разработки перетекает в параллельные друг другу процессы эксплуатации и

последующего сопровождения. На этапе сопровождения Исполнитель должен поддерживать работу продукта при возникновении несоответствий, проблем и вопросов. При возникновении такого фактора как ошибка Исполнитель обязан ее устранить.

Для соблюдения требования Заказчика по минимизации затрат в разработке будет использована реляционная система управления базами данных MySQL 8.0.18

MySQL является реляционной системой управления базами данных.

Реляционная система управления базами данных в свою очередь на данный момент представляют собой большинство проектов, реализованных с помощью информационных систем.

По данным исследований, проведенных исследовательской компанией International Data Corporation в 2009г. девяносто три процента проектов использует именно реляционный тип управления базами данных.

Основными конкурентами MySQL являются:

- oracle;
- IBM DB;
- microsoft SQL Server;
- линтер (от разработчика РЭЛЭКС);
- sybase.

Реляционная система управления базами данных MySQL имеет много плюсов:

- гибкость (поддержка многих типов таблиц, форматов чисел, строк и так далее);
- имеет принцип создания новых типов таблиц;
- открытая архитектура;
- бесплатна;
- быстрая и качественная поддержка;
- многопоточность;
- высокая эффективность;

- надежность;
- большое количество плагинов и расширений;
- простая установка;
- безопасность;
- универсальность;
- производительность.

Несмотря на большое количество плюсов, как и все системы управления базами данных MySQL также имеет свой ряд минусов. Она может уступать в надежности другим системам управления базами данных и имеет достаточно маленькую скорость разработки, но несмотря на это плюсы нивелируют все минусы этой системы.

В данном случае она удовлетворяет всем требованиям Заказчика (Безопасность, Надежность, Простота установки, Бесплатность).

Для поддержки MySQL будет использован программный продукт PHP.

PHP является интерпретируемым языком программирования, имеющим общее назначение, но в нашем случае будет использоваться для создания веб сайта в поддержку MySQL.

PHP на момент написания данной квалификационной работы является одним из самых распространённых и используемых языков программирования.

PHP стоит в одном ряду с таким гигантом как Java Server Pages и Perl от The Perl Foundation.

В рейтинге языков программирования, составленном компанией TIOBE в 2016г. PHP занял 6 место после Java, C, Python, C++, C#. В январе 2020 года PHP занимает 8 место в списке после опередивших его Visual Basic.NET и Java Script.

Основными плюсами PHP можно назвать:

- открытый доступ и как следствие бесплатная лицензия;
- маленькие время затраты на разработку;
- совместимость с большинством оперативных систем;

- отличная масштабируемость проектов;
- "прозрачность" кода;
- качественные поддержка и обновление;
- быстрая загрузка;
- экономичность.

PHP также как и MySQL удовлетворяет требованиям Заказчика в простоте использования и экономичности разработки.

3.3 Выбор системы управления базами данных информационной системы управления цепями поставок

На момент написания данной квалификационной работы самыми эффективными системами управления базами данных являются Oracle, MySQL и Microsoft SQL Server. На рисунке 17 изображён список самых популярный систем управления базами данных.

341 systems in ranking, December 2018

Rank			DBMS	Database Model	Score		
Dec 2018	Nov 2018	Dec 2017			Dec 2018	Nov 2018	Dec 2017
1.	1.	1.	Oracle +	Relational DBMS	1283.22	-17.89	-58.32
2.	2.	2.	MySQL +	Relational DBMS	1161.25	+1.36	-156.82
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1040.34	-11.21	-132.14
4.	4.	4.	PostgreSQL +	Relational DBMS	460.64	+20.39	+75.21
5.	5.	5.	MongoDB +	Document store	378.62	+9.14	+47.85
6.	6.	6.	IBM Db2 +	Relational DBMS	180.75	+0.87	-8.83
7.	7.	↑ 8.	Redis +	Key-value store	146.83	+2.66	+23.59
8.	8.	↑ 10.	Elasticsearch +	Search engine	144.70	+1.24	+24.92
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	139.51	+1.08	+13.63
10.	10.	↑ 11.	SQLite +	Relational DBMS	123.02	+0.31	+7.82

Рисунок 17 – Ранжирование систем управления базами данных

Рассмотрим более подробно наиболее популярные системы.

Первую позицию в данном списке занимает Oracle Database. Oracle Database имеет много преимуществ. Ключевыми преимуществами можно назвать: высокую производительность, большие базы данных,

самоуправление, простые средства разработки, масштабируемость. Все это обеспечивает ей такую популярность среди разработчиков.

MySQL является системой управления реляционными базами данных с открытым исходным кодом с моделью клиент-сервер.

Одна из причин почему MySQL так популярна среди пользователей это простота использования. MySQL позволяет изменить исходный код и для ее установки не требуется много времени.

Вторая причина — это высокая производительность. MySQL обеспечивает оптимальную скорость для эффективной работы в большинстве серверов. Эта система управления базами данных является очень безопасной и хорошо поддерживаемой разработчиками.

На третьей позиции находится Microsoft SQL Server. Основными преимуществами этой системы можно назвать:

- простая передача больших данных,
- интеграция данных,
- высокая производительность.

На основании рассмотрения наиболее популярных систем управления базами данных в качестве системы управления была выбрана MySQL.

3.4 Разработка физической модели данных информационной системы управления цепями поставок

После выбора системы управления базами данных для реализации проекта необходимо построить физическую модель данных. Физическая модель данных на логическом уровне представлена на рисунке 18.

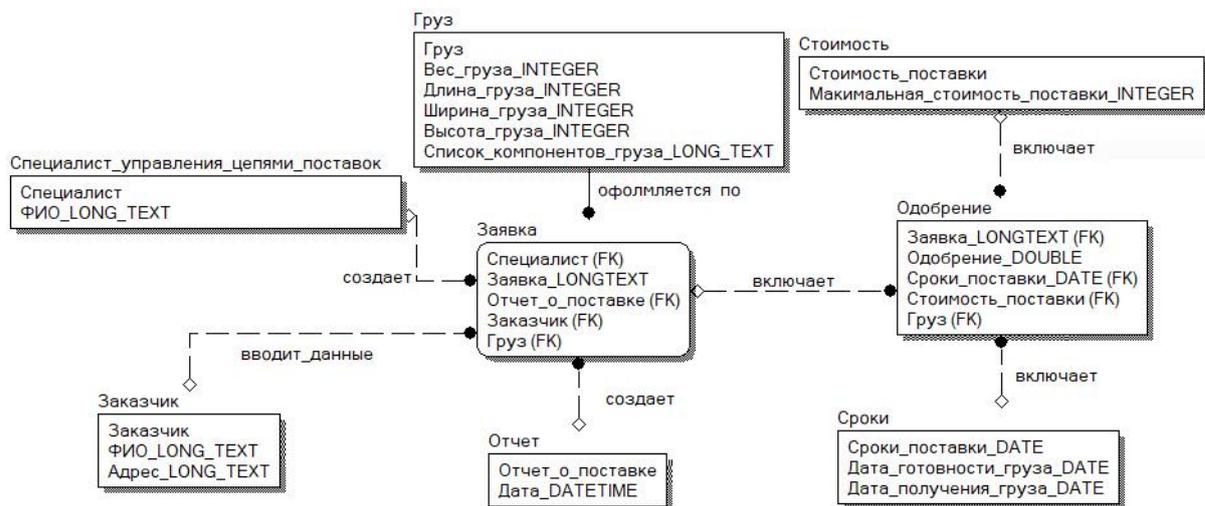


Рисунок 18 – Физическая модель данных

Физическая модель данных проекта построена с использованием методологии IDEF1X и представлена в двух формах, на логическом и физическом уровнях. Все связи на моделях не идентифицирующие.

3.5 Разработка программного обеспечения информационной системы управления цепями поставок

3.5.1 Схема взаимосвязи модулей приложения информационной системы управления цепями поставок

При работе конечный пользователь будет пользоваться веб сайтом, построенный в диалоговом режиме, то есть по мере того, как будет выполняться команды, сайт будет ожидать от конечного пользователя каких-либо действий.

Как только определенные действия будут выполняться, будет происходить считывание введенной информации и её обработка, запись данных в базу данных, преобразование информации и соответствующие расчёты, результаты работы будут выводиться на печать.

Технологические операции программного продукта:

- загрузка сайта,
- ввод данных,
- считывание данных,
- выполнение расчётов,
- вывод выходной информации в графической форме.

Программные модули автоматизированной информационной системы управления цепями поставок представлены на рисунке 19.

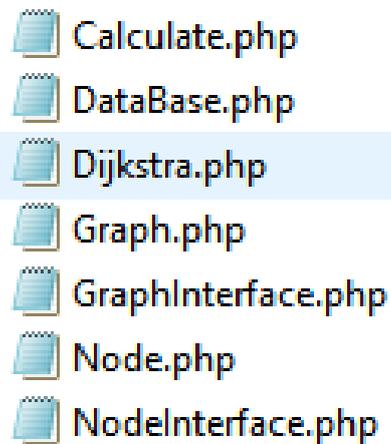


Рисунок 19 - Программные модули автоматизированной информационной системы управления цепями поставок

Все программные модули связаны между собой и связаны с главным модулем проекта. Взаимосвязь модулей представлена на рисунке 19.

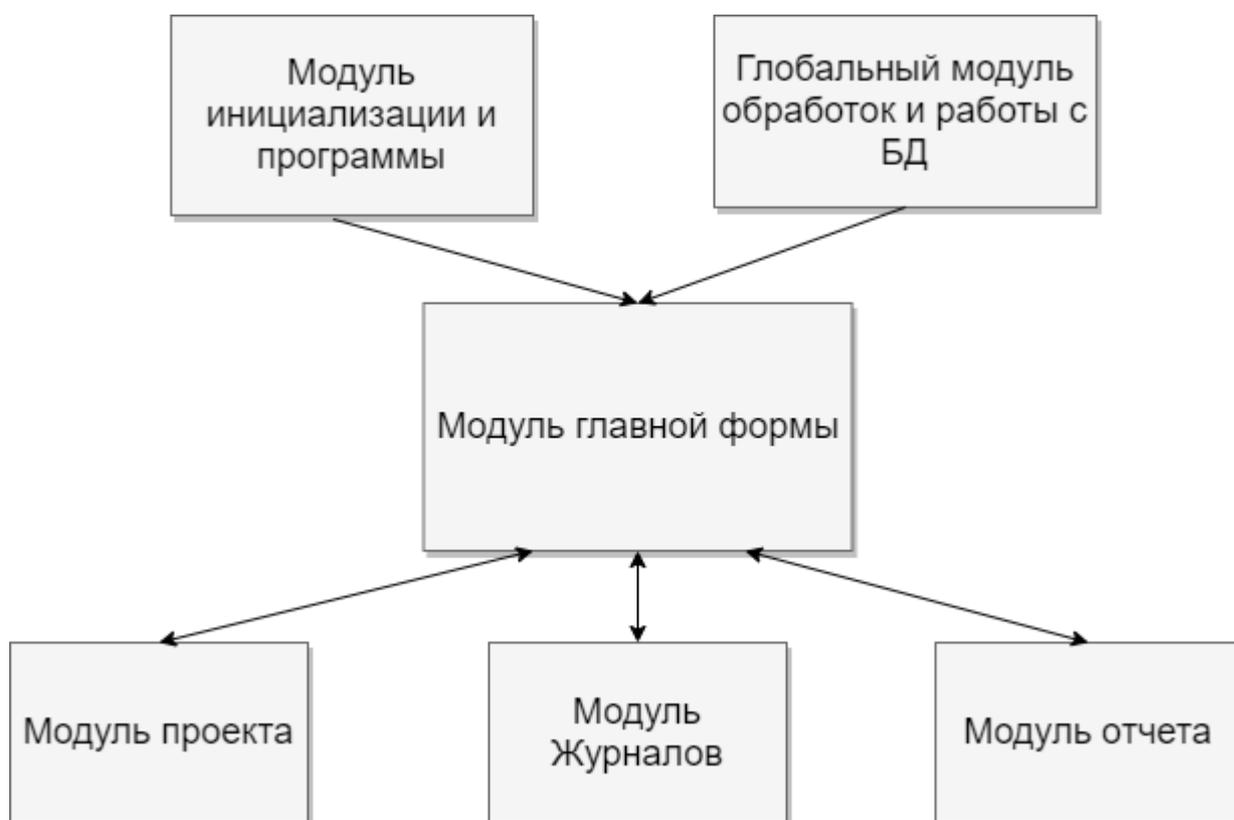


Рисунок 20 - Схема взаимосвязи модулей приложения автоматизированной информационной системы управления цепями поставок

Функциями информационной системы являются:

- внесение информации в таблицы информационной системы,
- поиск информации,
- обработка информации,
- вывод информации по результатам обработки.

После описания функций информационной системы необходимо перейти к описанию программных модулей автоматизированной информационной системы управления цепями поставок.

3.5.2 Описание модулей приложения информационной системы управления цепями поставок с примерами программного кода

На главной форме сайта можно получить доступ к заполнению формы на оформление заявки на расчет выгодной цепи поставки.

На данной форме необходимо заполнить обязательные поля, которые необходимы для корректной работы программы.

Обязательными являются поля:

- отправитель,
- получатель,
- дата поставки,
- описание и вес груза.

Для запуска процесса по расчёту выгодного пути доставки необходимо нажать «Расчитать стоимость».

Для расчёта стоимости перевозки подключается модуль Index. Фрагменты кода данного модуля представлены на рисунке 19.

```
<link rel="stylesheet" href="vendor/css/bootstrap.min.css"/>
<link rel="stylesheet" href="vendor/css/bootstrap-datetimepicker.min.css"/>

</head>
<body>
  <div class="container">
    <h1>Расчет стоимости перевозки груза</h1>

    <form action="calculate.php" method="post">
      <div class="row">
        <div class="col-md-6">
          <div class="form-group">
            <label for="from">Укажите отправителя:</label><br>
            <select class="form-control" id="from" name="from">
              <?=$options ?>
            </select>
          </div>
        </div>
        <div class="col-md-6">
          <div class="form-group">
            <label for="to">Укажите получателя:</label><br>
            <select class="form-control" id="to" name="to">
              <?=$options ?>
            </select>
          </div>
        </div>
      </div>

      <div class="row">
        <div class="col-md-6">
          <div class="form-group">
            <label for="datetimepicker">Укажите дату поставки:</label><br>
            <input type='text' class="form-control" name="date" id="datetimepicker"/>
          </div>
        </div>
        <div class="col-md-6">
          <div class="form-group">

```

Рисунок 21 - Фрагмент кода по расчету стоимости перевозки

Для корректного отображения полей интерфейса для заполнения используется модуль Calculate.

Фрагменты кода данного модуля представлены на рисунке 22.

```
<div class="header">
  <h2>Отчет</h2>
  <h3>Поручение Экспедитору/Перевозчику</h3>
</div>
<table class="table table-hover">
  <tbody>
    <tr>
      <th>Дата поставки:</th>
      <td><?= $_POST['date'] ?></td>
    </tr>
    <tr>
      <th>Заказчик/Клиент:</th>
      <td>000 "ВСР", РФ, 445000, Самарская область, г. Тольятти, ул. Борковская, 17.</td>
    </tr>
    <tr>
      <th>Экспедитор/Перевозчик:</th>
      <td><?= $calculate->getCarrierName() ?></td>
    </tr>
    <tr>
      <th>Наименование грузоотправителя:</th>
      <td><?= $calculate->from['name'] ?></td>
    </tr>
    <tr>
      <th>Адрес грузоотправителя:</th>
      <td><?= $calculate->from['address'] ?></td>
    </tr>
    <tr>
      <th>Наименование грузополучателя:</th>
      <td><?= $calculate->to['name'] ?></td>
    </tr>
    <tr>
      <th>Адрес грузополучателя:</th>
      <td><?= $calculate->to['address'] ?></td>
    </tr>
    <tr>
      <th>Информация о грузе:</th>
      <td><?= $_POST['description'] ?></td>
    </tr>
  </tbody>
</table>
```

Рисунок 21 - Фрагмент кода по отображению полей интерфейса

Исходный код программы представлен в приложении А.

3.6 Описание функциональности информационной системы управления цепями поставок

Автоматизированная информационная система по автоматизации процесса организации цепей поставок производственного предприятия предназначена для сокращения времени на разработку пути доставки груза при транспортировке.

Главная форма оформления заявки на расчет транспортировки изображена на рисунке 22.

Не защищено | super-project/index.php

Расчет стоимости перевозки груза

Укажите отправителя: Valeo Togliatti

Укажите получателя: Valeo Togliatti

Укажите дату поставки: 17.05.2020

Укажите размер груза:

Расчитать стоимость

май 2020						
пн	вт	ср	чт	пт	сб	вс
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Рисунок 22 – Главная форма оформления заявки

В поле «Укажите отправителя» необходимо выбрать отправителя из выпадающего списка среди существующих поставщиков предприятия, либо нажать в списке на «создать новый».

В поле «Укажите получателя» необходимо выбрать получателя из выпадающего списка среди существующих поставщиков предприятия, либо нажать в списке на «создать новый».

После указания отправителя и получателя необходимо указать дату поставки.

В поле «размер груза» необходимо указать длину, ширину, высоту и вес груза через пробел в соответствующем порядке. Длина, ширина и высота должна указываться в сантиметрах. Вес должен быть указан в килограммах.

После заполнения обязательных полей программа произведет расчёт наиболее выгодного пути доставки груза и выдаст отчет. Пример отчета изображен на рисунке 23.

Отчет
Поручение Экспедитору/Перевозчику

Дата поставки:	17.05.2020
Заказчик/Клиент:	ООО "BCP", РФ, 445000, Самарская область, г. Тольятти, ул. Борковская, 17.
Экспедитор/Перевозчик:	SCHENKER
Наименование грузоотправителя:	Valeo Togliatti
Адрес грузоотправителя:	Russia, Togliatti, Borkovskaya 17
Наименование грузополучателя:	Nidec Spain
Адрес грузополучателя:	Spain, STA PERPETUA DE MOGODA POLIGONO INDUSTRIAL CAN VINUYALS CALLE GUIFRE EL PILOS, S/N
Информация о грузе:	123
Путь:	Russia - Poland - Spain
Стоимость перевозки (включая экспортное оформление в стране отправления):	466

[Назад](#)

Рисунок 23 – Отчет о наиболее выгодном пути доставки

Программа учитывает расположение отправителя и получателя, вес груза, текущие тарифы с перевозчиками и выдает отчет с наиболее выгодной ценой доставки и скоростью перевозки.

3.7 Оценка и обоснование экономической эффективности разработки информационной системы автоматизации процесса по организации цепей поставок

3.7.1 Выбор методики расчета экономической эффективности

Экономическим эффектом автоматизированной информационной системы называют разницу между результатами деятельности системы, и произведенными затратами на разработку и внедрение системы [15, с. 45].

Основными источниками экономии в рамках предметной области являются:

- снижение трудоемкости;
- сокращение времени обработки информации;
- уменьшение численности персонала, выполняющего деятельность информационной системы;
- снижение затрат на расходные материалы;
- улучшение показателей деятельности в результате введения продукта по автоматизации.

Различают потенциальный и реальный экономический эффект.

Потенциальный экономический эффект может быть рассчитан до ввода продукта в эксплуатацию, по окончании разработки на основе достигнутых технико-экономических характеристик и прогнозных данных о максимальных объемах использования программного изделия [16, с. 322].

Для оценки экономической эффективности автоматизированной информационной системы используют стоимость затрат на внедрение системы, потенциальную экономию, окупаемость проекта. Перейдем к расчетам показателей экономической эффективности проекта.

3.7.2 Расчет показателей экономической эффективности проекта

Для того чтобы оценить экономическую эффективность разработки и внедрения автоматизированной информационной системы управления цепями поставок необходимо рассчитать показатели эффективности.

В данном проекте необходимо рассчитать прирост трудоемкости и стоимости.

Расчет прироста трудоемкости проекта представлен в таблице 5.

Таблица 5 – Расчет прироста трудоемкости проекта от внедрения новой системы

	Затраты в месяц		Абсолютный прирост	Рост	Прирост
	Текущие (час)	Планируемые (час)			
Трудоемкость	T ₀	T ₁	$\Delta T = T_1 - T_0$	$T_p = T_1 / T_0 \cdot 100\%$	$T_{пр} = T_p - 1$
	160	151	-9	94%	-6%

На основании данных, приведенных в таблице 5, получаем вывод о том, что трудоемкость при введении проекта снижается на шесть процентов.

Из данных прироста трудоемкости можно посчитать насколько изменится производительность труда по формуле 1.

$$П = \frac{1}{T_p} \quad (1)$$

где П – производительность труда;

T_p – рост.

$$П = 1 / 94\% = 1,0596\%$$

При расчёте получается, что производительность труда увеличится на 1,0596%.

Расчет прироста стоимости проекта представлен в таблице 6.

Таблица 6 – Расчет прироста стоимости проекта от внедрения новой системы

	Затраты на поставки в месяц		Абсолютный прирост	Рост	Прирост
	Текущие (руб.)	Планируемые (руб.)			
Стоимость	C_0	C_1	$\Delta C = C_1 - C_0$	$T_p = C_1 / C_0 * 100\%$	$T_{пр} = T_p - 1$
	117195	82115	-35080	70%	-30%

На основании данных, приведенных в таблице 6, получаем вывод о том, что стоимость затрат на поставки в месяц, при введении проекта, снижается на тридцать процентов.

Для расчётов трудоемкости необходимо рассчитать заработную плату исполнителя в час без учета накладных расходов, затраченных на проектирование, разработку, отладку и внедрение, тестирование, реализацию и сопровождение [19, с. 134]. Далее необходимо посчитать стоимость затрат на накладные расходы.

При расчёте трудоемкости и стоимости затрат будем учитывать погрешность в десять процентов

Расчет затрат на проектирование и внедрение проекта представлен в таблице 7

Таблица 7 – Расчет затрат на проектирование и внедрение проекта

Вид работ	Длительность (час)	Стоимость часа (руб.)	Затраты (руб.)
Проектирование	40	273	10920
Разработка	80	341	27300
Отладка и внедрение	25	239	5972
Тестирование	10	102	1024
Реализация	45	137	6143
Сопровождение	25	68	1706
Накладные расходы	0	0	31839
Итого			84903

Заработная плата программиста составляет в среднем на момент написания данной бакалаврской работы составляет 42000 рублей. Будем учитывать, что средняя стоимость отчислений работодателя в социальные фонды составляет тридцать процентов, средний рабочий день составляет восемь часов и количество рабочих дней в месяц – двадцать. Следовательно, заработная плата разработчика в час составляет 341 рубль.

В таблице для расчёта накладных расходов учитывается, что накладные расходы составлять шестьдесят процентов от общих затрат проекта.

Согласно подсчетам, выполненным в таблицах 5 и 6 затраты на внедрение проекта, составляют 84903 рублей, прибыль от внедрения проекта в месяц составляют 25080 рублей. Расчет экономии в год делается по формуле 2.

$$\text{Э} = \text{П} \cdot 12 \quad (2)$$

где Э – экономия;

П – производительность труда.

Следовательно, экономия проекта в год составляет 420964 рублей.

Посчитаем окупаемость проекта по формуле 3.

$$\text{О} = \frac{\text{З}}{\text{Э}} \cdot 12 \quad (3)$$

где О – окупаемость проекта;

З – затраты на внедрение;

Э – экономия.

Следовательно, окупаемость проекта составляет 2,42 месяца.

На основании полученных данных о стоимости затрат на разработку и внедрение проекта, полученных показателей эффективности, снижения трудоемкости, снижения стоимости на поставки, высокой экономии и окупаемости был сделан вывод о том, что разработка является экономически эффективной.

Выводы по главе 3

На первом этапе была выбрана архитектура автоматизированной информационной системы «клиент–сервер». В качестве системы управления была выбрана реляционная система управления базами данным MySQL.

Далее была разработана физическая модель информационной системы.

Для представления работы программного продукта была построена и описана схема взаимодействия модулей программы и была описана функциональность системы.

После построения схемы была произведена разработка программного обеспечения информационной системы и модулей её поддержки.

На заключительном этапе были проанализированы показатели экономической эффективности, стоимость затрат на внедрение и разработку программы и рассчитаны потенциальная экономия проекта и его окупаемость. Расчёты экономии и окупаемость проекта подтвердили экономическую эффективность внедрения продукта по автоматизации.

Заключение

В рамках данной работы было выполнено проектирование и разработка информационной системы по автоматизации процесса по организации цепей поставок производственного предприятия с использованием реляционной системы управления базами данных MySQL и языка программирования PHP.

Разработанная система снижает время затраты для определения необходимой цепи поставок и автоматизируется деятельность по расчёту необходимых тарифов, наиболее выгодного и быстрого пути доставки и наиболее низкой стоимости транспортировки.

Для достижения целей были проведены следующие этапы:

- проанализирована работа логистики и логистических потоков предприятия;
- рассмотрена деятельность специалистов по управлению цепями поставок и людей, непосредственно участвующих в цепи транспортировки;
- разработаны и проанализированы модели «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- определен путь перехода из одной модели в другую;
- описаны требования и задачи проектируемой системы;
- выполнено физическое моделирование на физическом и логическом уровнях;
- рассчитаны показатели экономической эффективности, экономия и окупаемость.

Контрольный пример автоматизированной информационной системы по автоматизации процесса по организации цепей поставок производственного предприятия был выполнен с помощью MySQL Community Server 8.0.18, PHP 7.2.25. и Wamp Server 3.0.0.

По расчетам экономической эффективности информационная система принесет снижение трудоемкости на шесть процентов и снижение стоимости

поставок на 30% в месяц. Окупаемость проекта составила 2,46 месяца. На основании сделанных расчетов был сделан вывод и принятии проекта по разработке информационной системы по автоматизации процесса по организации цепей поставок экономически эффективным.

Список используемой литературы и используемых источников

1. Аверченков В. И. Информационные системы в производстве и экономике : учебное пособие / В. И. Аверченков, Ф. Ю. Лозбинев, А.А.Тищенко – М. : Флинта, 2011 – 274 с.
2. Автоматизированные информационные системы в экономике: учебное пособие / Г. Г. Куликов, Е. А. Дронь, М. А. Шилина, Ю. О. Багаева: Уфимск. гос. авиац. техн. ун-т. – Уфа : УГАТУ, 2013. – 186 с.
3. Александров, Д. В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: учебное пособие / Д. В. Александров – М. : Финансы и статистика, 2012 – 225 с.
4. Блинов, А. О. Реинжиниринг бизнес-процессов: учеб.пособие / А. О.Блинов, О. С. Рудакова – М. : Юнити-Дана, 2012 – 342 с.86 с.
5. Вендров А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. - М. : Финансы и статистика, 2012. – 352 с.
6. Гвоздева, Т. В. Проектирование информационных систем: учебное пособие / Т. В. Гвоздева, Б. А. Баллод. – Ростов-н/Д : Феникс, 2014 – 508 с.
7. Должностная инструкция специалиста по управлению цепями поставок ООО «ВСП» [Рукопись]. – ООО «ВАЛЕО СЕРВИС РОССИЯ», 2014. – 5 с.
8. Заботина Н. Н. Проектирование информационных систем. Учебное пособие для студентов высших учебных заведений, обучающихся по специальности 080801 «Прикладная информатика (по областям) и другим экономическим специальностям» / Н. Н. Заботина – Москва : ИНФРА-М, 2013– 329 с.
9. Избачков, Ю. Информационные системы : учеб. для вузов / Ю. Избачков. – 3-е изд. – СПб. : Питер, 2013. – 544 с.

10. Ильин, В. В. Моделирование бизнес-процессов. Практический опыт разработчика / В. В.Ильин. – М. : «Вильямс», 2012. – 176 с.
11. Илюшечкин, В. М. Основы использования и проектирования баз данных : учеб. пособие / В. М. Илюшечкин. - М. : Юрайт, 2014 – 213 с.
12. Исаев Г. Н. Информационные технологии: учебное пособие / Г. Н. Исаев. – Изд-во : Омега-Л, 2012 – 464 с.
13. Косиненко Н. С. Информационные системы и технологии в экономике. Учебное пособие / Н. С. Косиненко, И. Г. Фризен - Москва : Дашкови К , 2012 - 303 с.
14. Кузнецов, С. Д. Базы данных. Модели и языки / С. Д. Кузнецов. – СПб. : Бином-Пресс, 2008. – 720 с.
15. Скрипкин, К. Г. Экономическая эффективность информационных систем / К. Г. Скрипкин. – М. : ДМК Пресс, 2012. – 256 с.
16. Трофимов, В. В. Информационные системы и технологии в экономике и управлении: учебник / В. В. Трофимов. – М. : Юрайт, 2011. – 521 с.
17. Устав ООО «ВСП» [Рукопись]. – ООО «ВАЛЕО СЕРВИС РОССИЯ», 2011. –245 с.
18. Харьков В. П. Информационные системы и технологии на предприятии. Учебное пособие / В. П. Харьков ; Нац. ин-т бизнеса. - Москва :Нац. ин-т бизнеса , 2012 - 210 с.
19. Царев, В. В. Оценка экономической эффективности инвестиций / В.В. Царев. – СПб. : Питер, 2013. – 464 с.
20. Чекмарев Ю. В. Автоматизация процессов обработки информации в статистике: учебное пособие / Ю. В. Чекмарев, И. А. Кашина, В. К. Кашин, Д.Ю. Нечаев. – Изд-во : ДМК Пресс, 2012 – 199 с.

Приложение А

Исходный код программы

```
<?php
require "models\Calculate.php";
$calculate = null;
$error = null;
try {
    $calculate = new Calculate($_POST['from'], $_POST['to']);
} catch (Exception $e) {
    $error = $e->getMessage();
}
?>
<!DOCTYPE html>
<html lang="ru">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Расчет стоимости перевозки груза</title>
        <link rel="stylesheet" href="vendor/css/bootstrap.min.css"/>
        <link rel="stylesheet" href="vendor/css/bootstrap-datetimepicker.min.css"/>
        <link rel="stylesheet" href="vendor/css/calculate.css"/>
    </head>
    <body>
        <div class="container report">
            <?php if ($error !== null): ?>
                <div class="header">
                    <h2>Ошибка!</h2>
                    <span><?= $error ?></span>
                </div>
            <?php else: ?>
                <div class="header">
                    <h2>Отчет</h2>
                    <h3>Поручение Экспедитору/Перевозчику</h3>
                </div>
                <table class="table table-hover">
                    <tbody>
                        <tr>
                            <th>Дата поставки:</th>
                            <td><?= $_POST['date'] ?></td>
                        </tr>
                        <tr>
                            <th>Заказчик/Клиент:</th>
                            <td>ООО "ВСП", РФ, 445000, Самарская область, г. Тольятти, ул.
Борковская, 17.</td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </body>
    </html>
```

```

        <th>Экспедитор/Перевозчик:</th>
        <td><?= $calculate->getCarrierName() ?></td>
    </tr>
    <tr>
        <th>Наименование грузоотправителя:</th>
        <td><?= $calculate->from['name'] ?></td>
    </tr>
    <tr>
        <th>Адрес грузоотправителя:</th>
        <td><?= $calculate->from['address'] ?></td>
    </tr>

    <tr>
        <th>Наименование грузополучателя:</th>
        <td><?= $calculate->to['name'] ?></td>
    </tr>
    <tr>
        <th>Адрес грузополучателя:</th>
        <td><?= $calculate->to['address'] ?></td>
    </tr>
    <tr>
        <th>Информация о грузе:</th>
        <td><?= $_POST['description'] ?></td>
    </tr>
    <tr>
        <th>Путь:</th>
        <td><?= $calculate->getCarrierPath() ?></td>
    </tr>
    <tr>
        <th>Стоимость перевозки (включая экспортное оформление в стране
отправления):</th>
        <td><?= $calculate->getCarrierPrice() ?> </td>
    </tr>

```

```

    </tbody>
</table>
<?php endif; ?>
<a href="index.php" class="btn btn-default pull-right">Назад</a>
</div>

```

```

<script src="vendor/js/bootstrap.min.js"></script>
</body>
</html>

```

```

<?php
require "models\DataBase.php";

```

```

$conn = DataBase::getConnection();

```

```

$sql = "SELECT id, name FROM provider";
$result = $conn->query($sql);

```

```

$options = "";

```

```

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $options .= '<option value="'. $row['id'] .'>'. $row['name'] .'</option>';
    }
}
?>
<!DOCTYPE html>
<html lang="ru">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Расчет стоимости перевозки груза</title>

        <link rel="stylesheet" href="vendor/css/bootstrap.min.css"/>
        <link rel="stylesheet" href="vendor/css/bootstrap-datetimepicker.min.css"/>
    </head>
    <body>
        <div class="container">
            <h1>Расчет стоимости перевозки груза</h1>
            <form action="calculate.php" method="post">
                <div class="row">
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="from">Укажите отправителя:</label><br>
                            <select class="form-control" id="from" name="from">
                                <?=$options ?>
                            </select>
                        </div>
                    </div>
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="to">Укажите получателя:</label><br>
                            <select class="form-control" id="to" name="to">
                                <?=$options ?>
                            </select>
                        </div>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="datetimepicker">Укажите дату поставки:</label><br>
                            <input type='text' class="form-control" name="date" id="datetimepicker"/>
                        </div>
                    </div>
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="description">Укажите размер груза:</label><br>
                            <input type='text' class="form-control" name="description" id="description"/>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </body>
</html>

```

```

        </div>

    </div>

    <button type="submit" class="btn btn-default pull-right">Рассчитать
стоимость</button>
    </form>
</div>
<script src="vendor/js/jquery-3.2.1.min.js"></script>
<script src="vendor/js/moment-with-locales.min.js"></script>
<script src="vendor/js/bootstrap.min.js"></script>
<script src="vendor/js/bootstrap-datetimepicker.min.js"></script>
<script type="text/javascript">
    $(function () {
        $('#datetimepicker').datetimepicker({
            locale: 'ru',
            stepping: 10,
            format: 'DD.MM.YYYY',
            defaultDate: new Date(),
        });
    });
</script>
</body>
</html>
<?php
require "models\Graph.php";
require "models\Node.php";
require "models\Dijkstra.php";
require "models\DataBase.php";
class Calculate
{
    public $from;
    public $to;
    public $carrier;
    /**
     * Соединение в базой
     * @var mysqli
     */
    private $_connection;
    /**
     * Calculate constructor.
     * @param $fromID
     * @param $toID
     * @throws Exception
     */
    public function __construct($fromID, $toID)
    {
        $this->_connection = DataBase::getConnection();
        $providers = $this->getDBProviders();
        if (empty($providers[$fromID]) || empty($providers[$toID])) {
            throw new \RuntimeException('Указан не существующий поставщик');
        }
    }
}

```

```

$this->from = $providers[$fromID];
$this->to = $providers[$toID];

$carriers = $this->getCarriers($this->from['country_id'], $this->to['country_id']);
$minCarrier = null;
if (!empty($carriers)) {
    $minCarrier = array_shift($carriers);
    foreach ($carriers as $carrier) {
        if ($carrier['price'] < $minCarrier['price']) {
            $minCarrier = $carrier;
        }
    }
}
$this->carrier = $minCarrier;
}
public function getCarrierName(){
    if ($this->carrier !== null) {
        return $this->carrier['name'];
    }
    return null;
}
public function getCarrierPrice(){
    if ($this->carrier !== null) {
        return $this->carrier['price'] . ' EUR';
    }
    return null;
}

public function getCarrierPath(){
    if ($this->carrier !== null) {
        return implode(' - ', $this->carrier['path']);
    }
    return null;
}
/**
 * получаю из базы всех поставщиков
 * @return array
 */
private function getDBProviders()
{
    $sql = "SELECT id, name, address, country_id FROM provider";
    $result = $this->_connection->query($sql);
    $providers = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $providers[$row['id']] = [
                'name' => $row['name'],
                'address' => $row['address'],
                'country_id' => $row['country_id'],
            ];
        }
    }
}

```

```

    return $providers;
}
/**
 * получаю из базы всех перевозчиков
 * @return array
 */
private function getDBCarriers()
{
    $sql = "SELECT id, name FROM carrier";
    $result = $this->_connection->query($sql);

    $carriers = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $carriers[$row['id']] = [
                'name' => $row['name'],
            ];
        }
    }
    return $carriers;
}

/**
 * получаю из базы всех перевозчиков
 * @return array
 */
private function getDBCcountry()
{
    $sql = "SELECT id, name FROM country";
    $result = $this->_connection->query($sql);
    $countries = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $countries[$row['id']] = [
                'name' => $row['name'],
            ];
        }
    }
    return $countries;
}

/**
 * получаю из базы все тарифы
 * @return array
 */
private function getDBTariffs()
{
    $sql = "SELECT id, carrier_id, country_deliver_id, country_receive_id, price FROM
tariff";
    $result = $this->_connection->query($sql);
    $tariffs = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {

```

```

        if (empty($tariffs[$row['carrier_id']])) {
            $tariffs[$row['carrier_id']] = [];
        }
        $tariffs[$row['carrier_id']][] = [
            'fromID' => $row['country_deliver_id'],
            'toID' => $row['country_receive_id'],
            'price' => $row['price'],
        ];
    }
}
return $tariffs;
}
/**
 * создаю графы на каждого перевозчика
 * @return array
 * @throws Exception
 */
private function getGraphs()
{
    $tariffs = $this->getDBTariffs();
    $graphs = [];
    foreach ($tariffs as $carrierID => $routes) {
        $graph = new Graph();
        foreach ($routes as $route) {
            $from = $route['fromID'];
            $to = $route['toID'];
            $price = $route['price'];
            if (!array_key_exists($from, $graph->getNodes())) {
                $from_node = new Node($from);
                $graph->add($from_node);
            } else {
                $from_node = $graph->getNode($from);
            }
            if (!array_key_exists($to, $graph->getNodes())) {
                $to_node = new Node($to);
                $graph->add($to_node);
            } else {
                $to_node = $graph->getNode($to);
            }
            $from_node->connect($to_node, $price);
        }
        $graphs[$carrierID] = $graph;
    }
    return $graphs;
}
/**
 * Получаю перевозчиков с путем и стоимостью
 * @param $fromID
 * @param $toID
 * @return array
 * @throws Exception
 */

```

```

private function getCarriers($fromID, $toID)
{
    $carriers = $this->getDBCarriers();
    $graphs = $this->getGraphs();
    $countries = $this->getDBCcountry();
    foreach ($graphs as $carrierID => $graph) {
        $startNode = $graph->getNode($fromID);
        $endNode = $graph->getNode($toID);
        $g = new Dijkstra($graph);
        $g->setStartingNode($startNode);
        $g->setEndingNode($endNode);
        $path = $g->getLiteralShortestPath();
        $price = $g->getDistance();
        $carriers[$carrierID]['price'] = $price;
        foreach ($path as $countryID) {
            $carriers[$carrierID]['path'][] = $countries[$countryID]['name'];
        }
    }
    // если стоимость пустая, то убираем перевозчика из выборки
    return array_filter($carriers, static function ($carrier) {
        return !empty($carrier['price']);
    });
}
}
<?php
class DataBase
{
    /**
     * @return mysqli
     */
    public static function getConnection(){
        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "graph";
        // Create connection
        $connection = new mysqli($servername, $username, $password, $dbname, 3309);
        // Check connection
        if ($connection->connect_error) {
            throw new \RuntimeException('Не удалось подключиться к базе: ' . $connection->connect_error);
        }
        return $connection;
    }
}
<?php
class Dijkstra
{
    protected $startingNode;
    protected $endingNode;
    protected $graph;
    protected $paths = array();
}

```

```

protected $solution = false;
/**
 * Создает новый алгоритм, требующий граф для работы.
 *
 * @param Graph $graph
 */
public function __construct(Graph $graph)
{
    $this->graph = $graph;
}
/**
 * Возвращает расстояние между начальной и конечной точкой.
 *
 * @return integer
 */
public function getDistance()
{
    if (!$this->isSolved()) {
        throw new \RuntimeException("Невозможно рассчитать расстояние нерешенного
алгоритма: \nВы забыли вызвать ->solve()?");
    }
    return $this->getEndingNode()->getPotential();
}
/**
 * Получает узел, на который мы указываем.
 *
 * @return Node
 */
public function getEndingNode()
{
    return $this->endingNode;
}
/**
 * Возвращает решение в удобочитаемом стиле.
 *
 * @return array
 * @throws Exception
 */
public function getLiteralShortestPath()
{
    $path = $this->solve();
    $literal = [];
    foreach ($path as $p) {
        $literal[] = $p->getId();
    }
    return $literal;
}
/**
 * Вычисляет кратчайший путь графа благодаря потенциалам, хранящимся в узлах.
 *
 * @return array
 */

```

```

public function getShortestPath()
{
    $path = array();
    $node = $this->getEndingNode();
    do {
        $path[] = $node;
        $node = $node->getPotentialFrom();
        if ($node === null) {
            break;
        }
    } while ($node->getId() != $this->getStartingNode()->getId());
    $path[] = $this->getStartingNode();
    return array_reverse($path);
}
/**
 * Получает узел, с которого мы начинаем, чтобы вычислить кратчайший путь.
 *
 * @return Node
 */
public function getStartingNode()
{
    return $this->startingNode;
}
/**
 * Устанавливает узел, на который мы указываем.
 *
 * @param Node $node
 */
public function setEndingNode(Node $node)
{
    $this->endingNode = $node;
}
/**
 * Устанавливает узел, с которого мы начинаем, чтобы вычислить кратчайший путь.
 *
 * @param Node $node
 */
public function setStartingNode(Node $node)
{
    $this->paths[] = array($node);
    $this->startingNode = $node;
}
/**
 * Решает алгоритм и возвращает кратчайший путь в виде массива.
 *
 * @return array
 * @throws Exception
 */
public function solve()
{
    if (!$this->getStartingNode() || !$this->getEndingNode()) {

```

```

        throw new \RuntimeException("Невозможно выполнить алгоритм без начальных и
конечных узлов");
    }
    $this->calculatePotentials($this->getStartingNode());
    $this->solution = $this->getShortestPath();
    return $this->solution;
}
/**
 * Рекурсивно вычисляет потенциалы графа, начиная с начальной точки,
 * которую вы указываете с помощью ->setStartingNode(), обходя граф по связям узла.
 *
 * @param Node $node
 * @throws Exception
 */
protected function calculatePotentials(Node $node)
{
    $connections = $node->getConnections();
    $sorted = array_flip($connections);
    krsort($sorted);
    foreach ($connections as $id => $distance) {
        $v = $this->getGraph()->getNode($id);
        $v->setPotential($node->getPotential() + $distance, $node);
        foreach ($this->getPaths() as $path) {
            $count = count($path);
            if ($path[$count - 1]->getId() === $node->getId()) {
                $this->paths[] = array_merge($path, array($v));
            }
        }
    }
    $node->markPassed();
    // Получить цикл через ближайшие соединения текущего узла
    // чтобы рассчитать их потенциалы.
    foreach ($sorted as $id) {
        $node = $this->getGraph()->getNode($id);
        if (!$node->isPassed()) {
            $this->calculatePotentials($node);
        }
    }
}
/**
 * Возвращает граф, связанный с этим экземпляром алгоритма.
 *
 * @return Graph
 */
protected function getGraph()
{
    return $this->graph;
}
/**
 * Возвращает возможные пути, зарегистрированные в графе.
 *
 * @return array

```

```

    */
protected function getPaths()
{
    return $this->paths;
}
/**
 * Проверяет алгоритм был решен или нет.
 *
 * @return boolean
 */
protected function isSolved()
{
    return ( bool )$this->solution;
}
}
<?php
require "models\GraphInterface.php";
class Graph implements GraphInterface
{
    /**
     * Все узлы на графе
     *
     * @var array
     */
    protected $nodes = array();
    /**
     * Добавляет новый узел в текущий граф.
     *
     * @param NodeInterface $node
     * @return Graph
     * @throws Exception
     */
    public function add(NodeInterface $node)
    {
        if (array_key_exists($node->getId(), $this->getNodes())) {
            throw new \RuntimeException('Невозможно вставить несколько узлов с одинаковым
идентификатором в граф');
        }
        $this->nodes[$node->getId()] = $node;
        return $this;
    }
    /**
     * Возвращает узел, идентифицированный с помощью $id, связанного с этим графом.
     *
     * @param mixed $id
     * @return Node
     * @throws Exception
     */
    public function getNode($id)
    {
        $nodes = $this->getNodes();
        if (!array_key_exists($id, $nodes)) {

```

```

        throw new \RuntimeException("Невозможно найти { $id } на графе");
    }
    return $nodes[$id];
}
/**
 * Возвращает все узлы, которые принадлежат этому графу.
 *
 * @return array
 */
public function getNodes()
{
    return $this->nodes;
}
}
<?php
interface GraphInterface
{
    /**
     * Добавляет новый узел в текущий граф.
     *
     * @param NodeInterface $node
     * @return Graph
     */
    public function add(NodeInterface $node);
    /**
     * Возвращает узел, идентифицированный с помощью $id, связанного с этим графом.
     *
     * @param mixed $id
     * @return Node
     * @throws Exception
     */
    public function getNode($id);
    /**
     * Возвращает все узлы, которые принадлежат этому графу.
     *
     * @return array
     */
    public function getNodes();
}
<?php
require "models\NodeInterface.php";
class Node implements NodeInterface
{
    protected $id;
    protected $potential;
    protected $potentialFrom;
    protected $connections = array();
    protected $passed = false;
    /**
     * Создает новый узел, требующий ID, чтобы избежать коллизий.
     *
     * @param mixed $id

```

```

*/
public function __construct($id)
{
    $this->id = $id;
}
/**
 * Соединяет узел с другим $node.
 * Можно указать $distance, чтобы сбалансировать соединение.
 *
 * @param NodeInterface $node
 * @param integer $distance
 */
public function connect(NodeInterface $node, $distance = 1)
{
    $this->connections[$node->getId()] = $distance;
}
/**
 * Возвращает расстояние до узла.
 *
 * @param NodeInterface $node
 * @return array
 */
public function getDistance(NodeInterface $node)
{
    return $this->connections[$node->getId()];
}
/**
 * Возвращает соединения текущего узла.
 *
 * @return array
 */
public function getConnections()
{
    return $this->connections;
}
/**
 * Возвращает идентификатор этого узла.
 *
 * @return mixed
 */
public function getId()
{
    return $this->id;
}
/**
 * Возвращает потенциал узла.
 *
 * @return integer
 */
public function getPotential()
{
    return $this->potential;
}

```

```

/**
 * Возвращает узел, который дал текущему узлу свой потенциал.
 *
 * @return Node
 */
public function getPotentialFrom()
{
    return $this->potentialFrom;
}
/**
 * Возвращает, прошел узел или нет.
 *
 * @return boolean
 */
public function isPassed()
{
    return $this->passed;
}
/**
 * Отмечает этот узел как пройденный, что означает, что в
 * области видимости он уже обработан для вычисления его потенциала.
 */
public function markPassed()
{
    $this->passed = true;
}
/**
 * Устанавливает потенциал для узла, если узел не имеет потенциала
 * или тот, который он имеет, выше, чем новый.
 *
 * @param integer $potential
 * @param NodeInterface $from
 * @return boolean
 */
public function setPotential($potential, NodeInterface $from)
{
    $potential = ( int )$potential;
    if (!$this->getPotential() || $potential < $this->getPotential()) {
        $this->potential = $potential;
        $this->potentialFrom = $from;
        return true;
    }
    return false;
}
}
<?php

interface NodeInterface
{
    /**
     * Соединяет узел с другим $node.
     * Можно указать $distance, чтобы сбалансировать соединение.

```

```

*
* @param NodeInterface $node
* @param integer $distance
*/
public function connect(NodeInterface $node, $distance = 1);
/**
* Возвращает соединения текущего узла.
*
* @return array
*/
public function getConnections();
/**
* @return mixed
*/
public function getId();
/**
* Возвращает потенциал узла.
*
* @return integer
*/
public function getPotential();
/**
* Возвращает узел, который дал текущему узлу свой потенциал.
*
* @return Node
*/
public function getPotentialFrom();
/**
* Возвращает, прошел узел или нет.
*
* @return boolean
*/
public function isPassed();
/**
* Отмечает этот узел как пройденный, что означает, что в
* области видимости он уже обработан для вычисления его потенциала.
*/
public function markPassed();
/**
* Устанавливает потенциал для узла, если узел не имеет потенциала
* или тот, который он имеет, выше, чем новый.
*
* @param integer $potential
* @param NodeInterface $from
* @return boolean
*/
public function setPotential($potential, NodeInterface $from);
}

```