Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский университет   
«Высшая школа экономики»

###### Факультет: бизнес-информатики

###### Кафедра: Корпоративных информационных систем

###### ВЫПУСКНАЯКВАЛИФИКАЦИОННАЯ РАБОТА

На тему “Модернизация бизнес-процесса логистических поставок в информационной системе”

Студент группы № 472

Пичугин Дмитрий Андреевич

Руководитель ВКР

Доцент кафедры КИС, к.т.н. Моргунов Александр Федорович

Москва, 2013

Оглавление

[Введение 4](#_Toc357594663)

[1.1 Бизнес-процессы компании X на момент до введения системы 6](#_Toc357594664)

[1.1.1 Бизнес-процесс заказа и доставки ингредиентов на склад 6](#_Toc357594665)

[1.1.2 Бизнес-процесс доставки заказов покупателям 8](#_Toc357594666)

[1.1.3 Степень автоматизации БП на момент «до введения системы» 10](#_Toc357594667)

[1.2 Обзор логистического блока ERP системы 10](#_Toc357594668)

[1.2.1 1С:Предприятие 8. TMS Логистика. Управление перевозками 12](#_Toc357594669)

[1.2.2 1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS 12](#_Toc357594670)

[1.3 Постановка задачи 15](#_Toc357594671)

[2.1 Описание программы 18](#_Toc357594672)

[2.1.1 Сервер 18](#_Toc357594673)

[2.1.1.1 Функции 18](#_Toc357594674)

[2.1.1.2 Схема взаимодействия классов и объектов структуры сервера. 20](#_Toc357594675)

[2.1.2 Клиент на Android 22](#_Toc357594676)

[2.1.2.1 Функции 22](#_Toc357594677)

[2.1.2.2Схема взаимодействия классов и структурных объектов программы 23](#_Toc357594678)

[2.2 Схема базы данных 24](#_Toc357594679)

[2.2.1 Список таблиц базы: 24](#_Toc357594680)

[2.2.2 Описание структуры данных 26](#_Toc357594681)

[2.2.2.1 Table\_Orders: 26](#_Toc357594682)

[2.2.2.2 Table\_MealsList: 27](#_Toc357594683)

[2.2.2.3 Table\_Customers: 27](#_Toc357594684)

[2.2.2.4 Table\_Meals: 28](#_Toc357594685)

[2.2.2.5 Table\_Recipes: 29](#_Toc357594686)

[2.2.2.6 Table\_Measurment 30](#_Toc357594687)

[2.2.2.7 Table\_Ingredients 30](#_Toc357594688)

[2.2.2.8 Table\_Batches 31](#_Toc357594689)

[2.2.2.9 Table\_Locations: 32](#_Toc357594690)

[2.2.2.10 Table\_Suppliers: 33](#_Toc357594691)

[2.2.2.11 Table\_Status: 33](#_Toc357594692)

[2.2.2.12 Table\_SuppliersIngredientsList: 34](#_Toc357594693)

[2.2.2.13 Table\_Cars: 35](#_Toc357594694)

[2.2.2.14 Table\_Overseers: 36](#_Toc357594695)

[2.2.2.15 Table\_Drivers: 36](#_Toc357594696)

[2.2.3 Использование структуры данных в ИС 37](#_Toc357594697)

[3.1 Руководство пользователя 39](#_Toc357594698)

[3.1.2 Схема взаимодействия форм в приложении 39](#_Toc357594699)

[3.1.3 Этапы процесса 39](#_Toc357594700)

[3.1.3.1 Аутентификация и вход в систему 40](#_Toc357594701)

[3.1.3.2 Ввод заказа в базу и ожидание его готовности 40](#_Toc357594702)

[3.1.3.3 Назначение заказа на машину 43](#_Toc357594703)

[Заключение 45](#_Toc357594704)

[Список литературы: 47](#_Toc357594705)

[Список сокращений: 48](#_Toc357594706)

# Введение

В своей работе я сосредоточусь на описании возможного улучшения бизнес-процесса (БП) логистики ресторана, занимающегося розничной доставкой свежеприготовленной еды. Ресторан принимает заказы посредством телефонных звонков, либо через веб-сайт. Компания предоставляет на выбор широкий спектр блюд и гарантирует быструю доставку заказа в пределах города.

Компания заказывает ингредиенты у нескольких проверенных поставщиков по мере необходимости и хранит свои запасы на складе, этот процесс находиться в ведении учетного отдела компании. Обработку заявок через сайт и телефонные звонки клиентов осуществляет отдел продаж, после он переправляет их на транспортный отдел. Отдел транспорта имеет ограниченный парк автомобилей и осуществляет доставку заказа покупателям, а так же перевозку ингредиентов от поставщиков и доставку их на склад, принадлежащий компании.

Сейчас в данной компании бизнес-процесс приводится в действие в основном с помощью бумажных документов и отдельных информационных систем, облегчающих выполнение и ускорение данных процессов, что оставляет большой простор для улучшения процессов компании путем унификации и сведения всех разнородных источников данных в одну стабильную и удобную систему, что позволит ускорить выполнение процессов и обеспечит дальнейшую возможность к информатизации данного предприятия.

Цель работы состоит в совершенствовании выполнения бизнес-процесса логистических поставок путем доработки информационной системы, объединяющей в себе весь необходимый для функционирования БП инструментарий.

Задачи исследования:

1. Проектирование структуры и логики ИС, улучшающей выполнение БП.
2. Разработка и проектирование базы данных для ИС.
3. Создание основных форм приложения.

В первой главе рассмотрен БП компании в состоянии «до введения системы» и опишу функционал уже созданных программных решений, потенциально способных решить эту задачу на примере 1С: Enterprise.

Вторая глава будет посвящена описанию структуры программного решения и разработке схему базы данных для хранения информации, необходимой предприятию.

В третьей главе будет предоставлено краткое руководство пользователя в качестве примера работы с приложением.

# 1.1 Бизнес-процессы компании X на момент до введения системы

В моей работе я буду рассматривать два бизнес-процесса логистики данной компании, абстрагируясь от всего остального. Упрощенный вариант схемы обоих БП представлен на рисунках 1 и 2:

## 1.1.1 Бизнес-процесс заказа и доставки ингредиентов на склад

БП заказа и доставки ингредиентов на склад инициируется после запроса другого отдела, чаще всего отдела по продажам, при отсутствии на складе необходимых товаров. Процесс состоит из следующих этапов:

1. Получение заявки от одного из отделов на заказ товаров у поставщиков.
2. Выбор поставщика из списка (фирма уже имеет список доверенных поставщиков, с которыми она работала ранее) в зависимости от объема и срочности заказа и отправка заявки.
3. Получение ответа на заявку с указание сроков поставки.
4. Получение уведомления о готовности заказа.
5. Учетный отдел пересылает транспортному отделу данные о заказе. Транспортный отдел отправляет свободную машину забрать заказ.
6. После доставки заказа на склад транспортный отдел отправляет в учетный отдел уведомление о доставке заказа
7. После получения уведомления о доставке заказа, учетный отдел вносит соответствующие данные в систему. На этом процесс завершается.

Этот процесс носит упрощенный характер, но иллюстрирует схему работы компании достаточно детально, чтобы предоставить общее понимание процесса её работы.



Рисунок 1 Бизнес-процесс заказа и доставки ингредиентов на склад

1.1.2 Бизнес-процесс доставки заказов покупателям

БП доставки заказов покупателю – один из самых критичных для компаний, занимающихся интернет продажами. От скорости и точности его выполнения будет зависеть репутация и доверие покупателей к ресторану, что вынуждает компанию тратить немалые средства на поддержание безукоризненной четкости и достаточной скорости выполнения данного процесса. Этот процесс сложнее предыдущего и состоит из следующих этапов:

1. Получение заказа от пользователя путём телефонного звонка, либо заполнения формы заказа на веб-сайте.
2. Верификация заказа и проверка наличия ингредиентов для блюда на складе. В случае наличия, процесс передается дальше по цепочке, в ином случае процесс завершается и начинается инициализация БП заказа и доставки ингредиентов на склад
3. Заказ перенаправляется на транспортный отдел.
4. Транспортный отдел отправляет заявку на получения блюда со склада в отдел учёта.
5. Отдел учета верифицирует заявку и возвращает подтверждение в транспортный отдел.
6. Транспортный отдел ищет свободную машину и получает блюда со склада.
7. Машина доставляет товар покупателю/покупателям.
8. После доставки товара транспортный отдел пересылает данные о выполненной заявке в учетный отдел.
9. Учетный отдел вносит данные в систему. На этом этапе процесс завершается.



Рисунок 2 Бизнес-процесс доставки заказов покупателям

Это БП требует продуманной логики и огромного количества своевременно полученной информации для выполнения, поскольку любое промедление будет подрывать доверие к ресторану и ухудшать вкус еды, а, следовательно, и удовлетворенность покупателей работой компании.

## 1.1.3 Степень автоматизации БП на момент «до введения системы»

На данный момент документооборот компании в рамках данных БП включает в основном бумажные документы и отдельные информационные системы, контролирующие некоторые стадии бизнес-процесса (сайт, через который делаются заказы; внутренняя система учета бухгалтерии). Документы не приведены к одному общему стандарту и требуется немало времени на загрузку документов в существующие локальные ИС и выгрузку информации из них.

Диспетчеры, контролирующие машины доставки, не имеют централизованной связи с водителями и не могут полноценно контролировать их прогресс и местоположение (переназначать заказы, планировать маршрут с учетом препятствий, получать информацию о прогрессе от водителей).

Заказы на ингредиенты посылаются только после запроса со стороны поваров и отправляются компании-поставщику с помощью курьера, что требует примерно день на всю процедуру заказа. Ведение статистики, позволяющей построить модель потребности в ингредиентах для ресторана, отсутствует.

Все это оставляет огромные возможности для улучшения бизнес-процесса путем его информатизации.

# 1.2 Обзор логистического блока ERP системы

Для этого исследования я выбрал систему «1С:Предприятие 8».

**1С:Предприятие** — программный продукт компании 1С, предназначенный для автоматизации деятельности на предприятии. По информации IDC на российском рынке в 2010 году «1С:Предприятие » занимала 26% рынка, в основном позиционируя себя как простую и надежную ERP-систему для среднего бизнеса.

Система программ «1С:Предприятие 8» включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.

Гибкость платформы позволяет применять 1С:Предприятие 8 в самых разнообразных областях:

* автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.
* поддержка оперативного управления предприятием;
* автоматизация организационной и хозяйственной деятельности;
* ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность;
* широкие возможности для управленческого учета и построения аналитической отчетности, поддержка многовалютного учета;
* решение задач планирования, бюджетирования и финансового анализа;
* расчет зарплаты и управление персоналом;
* другие области применения.

В рамках этого исследования нас будет интересовать лишь узкий набор из двух конфигураций, наиболее подходящих для решения задач описанных выше бизнес-процессов ресторана.

## 1.2.1 1С:Предприятие 8. TMS Логистика. Управление перевозками

Система управления перевозками "1С:Предприятие 8. TMS Логистика. Управление перевозками" ориентирована, прежде всего, на Компании, которым в процессе осуществления своей деятельности требуется решение задач транспортной логистики.

Функциональность конфигурации "1С:TMS Логистика. Управление перевозками" определяется списком подсистем, которые входят в ее состав:

* управление нормативно‑справочной информацией
* управление потребностями в перевозке грузов
* управление заданиями на перевозку грузов
* формирование рейсов
* управление ресурсами для обеспечения рейсов
* контроль за выполнением рейсов
* управление тарифной политикой Компании
* управление взаимодействиями
* управление доступом
* получение аналитической отчетности
* визуализация информации на электронных картах.

## 1.2.2 1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS

"1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS" - новое специализированное решение на платформе "1С:Предприятие 8.2", предназначенное для осуществления ГЛОНАСС/GPS мониторинга подвижных объектов, транспорта и персонала. Функциональные возможности решения позволяют получать в реальном времени информацию о перемещении объектов, на которых установлены автомобильные и персональные трекеры, GPS-навигаторы с модулем GSM, КПК и коммуникаторы на базе WindowsMobile 5.0 и выше с GPS и GPRS модулями. Возможно получение информации с различных датчиков (температурный, датчик уровня топлива, тревожная кнопка SOS, CAN шина и др.).

Программный продукт "1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS" обеспечивает следующие возможности:

* Осуществление контроля местоположения и фактического пробега объекта;
* Формирование отчетов по фактическому расходу топлива и возможность фиксировать все заправки и сливы;
* Оповещения при наступлении настроенных событий;
* Пресечение отклонений от запланированных маршрутов;
* Получение сигнала от тревожной кнопки SOS для обеспечения безопасности перевозок;
* Контроль скоростного режима, превышения допустимой скорости;
* Возможность составления отчета как по одной единице техники, так и по всему автопарку включая арендованный транспорт;
* Контроль работы пеших сотрудников.

"1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS" при необходимости может быть интегрирован как в типовые, так и в отраслевые решения на платформе "1С:Предприятие 8".

Система управления перевозками «1С:Предприятие 8. TMS Логистика. Управление перевозками» предоставляет основной функционал для управления логистикой, все жизненно важный функции для контроля и учета перевозок. Дополненная же «1С:Предприятие 8. Центр спутникового мониторинга ГЛОНАСС/GPS» она будет предоставлять всё необходимое для четкого мониторинга статусов всех доставок блюд клиентам. Для ресторана самым важным будет:

1. Возможность регистрации предварительной потребности в перевозке грузов и контроля её последующей обработки.
2. Возможность регистрации Заданий на перевозку грузов и контроля исполнения этих заданий.
3. Возможность фиксирования факта прохождения точек маршрута
4. Возможность контроля за процессом выполнения рейсов, включая состояние машины и постоянный мониторинг за её местоположением
5. **Визуализация информации на электронных картах**

**В целом система предоставляет всё необходимое для контроля за логистическим процессом ресторана, но она не учитывает некоторых возможностей бизнеса, которые при их появлении в системе могли бы стать важным подспорьем для ресторана, улучшив скорость и стабильность доставки. Так же можно интегрировать в систему некоторые недавно появившиеся сервисы для улучшения качества и возможностей системы, например, «Яндекс-пробки», сервис, который уже сейчас способен предсказывать образование пробок за час, что может быть критично для скорости доставки еды.**

# **1.3 Постановка задачи**

**В своей** р**аботе я планирую рассмотреть лишь некоторые фрагменты БП, которые** я **считаю можно улучшить путем доработки программного обеспечения (ПО), которое добавит новый функционал.**

В первую очередь необходимо доработать системы мониторинга положения машин и визуального отображения их на карте. «1С:Предприятие» способно выводить расположение машин на карту, но необходимо глубже интегрировать эти возможности с новыми разработками и некоторым комбинированием с другими системами.

1. **Включить в составление маршрутов для автомобилей такой сервис, как «Яндекс-Пробки». Думаю, можно не говорить, что в больших городах (например, Москве) пробки являются важным временным фактором для любой компании связанной с доставкой, особенно для ресторанов, развозящих по заказу еду. Любое промедление ухудшает сервис, поэтому необходимо сделать всё, чтобы уменьшить риск попадания в пробку. «Яндекс-Пробки» могут предоставить необходимую информацию, на основе которой можно рассчитать путь, свободный от пробок.**
2. **Сейчас система может визуализировать положение автомобилей и их состояние, что упрощает контроль над ними. Но можно также в реальном времени визуализировать каждый новый заказ, поступивший в базу. Это упростит составление более коротких и эффективных маршрутов для диспетчеров и частично решит проблему отмены заказов, потому что можно будет найти клиента, который заказывал точно такое же блюда в радиусе досягаемости машины.**

**Во-вторых, необходимо упрощение БП путем интеграции разных систем друг с другом и в результате – уменьшение количества стадий в процессе. В БП доставки заказов есть пункт отправки заявок в учетный отдел, чтобы удостовериться в поступлении блюд для доставки на склад (рисунок 3). Это занимает время и требует выполнения множества однотипных операций. Намного проще выводить информацию о готовых к отправке блюдах на экран диспетчера, вкупе с информацией о месте их назначения. Это позволит упростить БП, убрав из него взаимодействие с учетным отделом на этой стадии.**

**.**

Рисунок 3 Отправка заявок в учетный отдел

В-третьих, требуется увеличить скорость выполнения БП заказа и доставки ингредиентов путем внедрения у поставщиков специальной системы слежения за статусом выполнения заказов и плотной интеграции этой системы с системой-приемником у ресторана. После подачи заявки на ингредиенты ресторан вынужден ждать, пока заказ будет выполнен и придет оповещение от поставщика о возможности вывоза ингредиентов (рисунок 4), что может занять время, критичное для ресторана. В случае внедрения подобной системы ресторан сможет постоянно отслеживать статус своих заказов и отправлять машины за ингредиентами точно в срок, минуя возможные задержки.



Рисунок 4 Подача заявки

Для того чтобы достичь этой цели в первую очередь я планирую создать собственную простую систему слежения за объектами, которая сможет выполнять все вышеперечисленные улучшения. Она будет состоять из сервера, на котором будет располагаться основная часть ПО. В неё будут входить компоненты для слежения за машинами и визуализации информации о заказах, компонент для отслеживания статуса заказов по ингредиентам.

Устройств на Android OS, используемых для слежения за машинами и постоянной коммуникации с водителями. На этих устройствах будет программа-клиент, которая позволит отслеживать появление новых заказов, отображать карту и уведомления для водителей, так же оно будет предоставлять обратную связь с диспетчером.

# 2.1 Описание программы

Программная реализация будет состоять из двух частей: сервера, расположенного на веб-хостинге и клиентское приложение, написанной под Android, которая устанавливается на планшетный компьютер.

## 2.1.1 Сервер

Сервер предоставляет основной функционал ИС, именно на нем и будет происходить вводу данных, их обработка и дальнейшая работа с ними. Сервер состоит из двух подпрограмм ориентированных на разные, пусть и взаимосвязанные задачи: на работу с поставщиками ингредиентов и на работу с клиентами.

Интерфейс работы с поставщиками предназначен для диспетчеров, которые находятся в постоянном взаимодействии с поставщиками и должны за минимальное время получить информацию о недостатке определенных ингредиентов от поваров или диспетчеров, отвечающих за связь с клиентами, и быстро сформировать и отправить заказ поставщику.

Часть, отвечающая за работу с клиентами, упрощает ввод данных о поступивших заказах, а так же предоставляет возможность мониторинга их состояния, вывода в отдельную форму заказов, подготовленных к транспортировке. Так же в ней присутствует интегрированный в приложение веб-сайт с встроенным функционалом Google Maps, который позволяют визуально планировать маршрут, одновременно получая информацию о положение курьеров и любых препятствиях на пути.

### 2.1.1.1 Функции

Функции, который должен выполнять сервер для полноценной работы информационной системы:

1. Хранение информации в базе данных, поднятой на веб-хостинге (база будет работать на MS SQL)
   1. Хранение информации о бывших клиентах и их местоположении.
   2. Хранение информации об ингредиентах и их наличии на складе.
   3. Хранение библиотеки блюд и их рецептов.
   4. Хранение данных о поставщиках и степени доверия к ним.
   5. Хранение информации о текущих заказах.
   6. Хранение информации о текущих заявках на ингредиенты
   7. Изменение, дополнение и удаление информации, хранящейся в БД
   8. Визуализация информации о поступлении заказов на Google Maps, встроенных в форму сервера
2. Автоматический мониторинг необходимости в ингредиентах и оповещение о необходимости заказа
3. Создание заказа поставщикам и автоматический мониторинг его состояния
4. Контроль за местоположением транспортных средств и поддержание связи с водителями
5. Расчет оптимального маршрута для каждого транспортного средства с учетом пробок и прочих препятствий (реализовать с помощью сервисов Google Maps)
6. Прогнозирование потребности ресторана в ингредиентах с целью ускорить выполнение заказов путем выполнение БП заказа ингредиентов до, а не после поступления заказа
7. Мониторинг ингредиентов на складе по сроку годности
8. Возможность для клиента отменить заказ, система должна иметь возможность оперативно сообщить поварам об отмене, либо, если блюдо уже готово, спроектировать новый маршрут для транспортного средства с целью доставки (при условии наличия такого же заказа у другого клиента) блюда в кратчайшие сроки.

### 2.1.1.2 Схема взаимодействия классов и объектов структуры сервера.



Рисунок 5. Схема классов сервера

Список классов и объектов:

1. Login.cs – класс формы, отвечающей за аутентификацию пользователей и их вход в нужную часть системы.
2. Customers.cs – класс формы, отвечающей за прием заказов от клиентов, обработку и распределению их по машинам. Так же отвечает за слежение за машинами
3. Suppliers.cs - класс формы, отвечающей за взаимодействие с поставщиками и мониторинг как заказов поставщикам, так и запросов производственных подразделений.
4. CustomerOrderWork.cs – класс формы, отвечающий за добавление\редактирование заказа клиента
5. OrderWork.cs – класс формы, отвечающий за добавление\редактирование заказа поставщику на поставку ингредиентов
6. CustomersAdd.cs – класс формы, отвечающий за добавление нового клиента в базу системы
7. MealsChoose.cs – класс формы, отвечающий создание списка блюд, заказанных клиентом
8. SuppliersAdd.cs – класс формы, отвечающий за добавление нового поставщика в систему
9. ExecuteSQL.cs – класс, ответственный за работу с основной базой данных. Является посредником между БД и остальными классами приложения. Выполняет посланную в него команду и возвращает результирующий набор.
10. ThreadWorking.cs – класс, отвечающий за работу с потоками в рамках приложения(каждый пользователь работает в своем потоке, некоторые долгие операции так же выполняются в отдельном потоке для того, чтобы можно было одновременно выполнять несколько операций)
11. GlobalVariables.cs – класс, хранящий в себе все глобальные переменные, используемые в разных классах приложения
12. User.cs – класс, создающий экземпляр для каждого пользователя и хранящий в себе данные о пользователе для использования в рамках приложения.
13. Website – веб-сайт, несущий в себе функционал для работы в Google Maps. Встраивается в форму Customers.
14. Database – основная БД, хранящая в себе данные, используемые в приложении. Базируется на веб-хостинге для обеспечения удаленного доступа к базе.

## 2.1.2 Клиент на Android

Программа предназначена для использования курьерами компании и предоставляет доступ к информации по маршруту, которая хранится на сервере для защиты от неавторизованного доступа. Так же приложение предоставляет обратную связь с диспетчером и используется для отслеживания положения курьера на карте города.

### 2.1.2.1 Функции

Функции программы клиента на Android:

1. Агрегировать информацию о назначенных на транспортное средство заказах из БД и загружать её во внутреннюю память устройства
2. Визуализировать информацию о назначенных заказах на Google Maps, в том числе и проложенный диспетчером маршрут
3. Проверять обновления базы заказов в автоматическом режиме при наличии подключения к интернету
4. Наличие обратной связи с сервером, позволяющей изменять статус заказов
5. Недвусмысленно уведомлять водителя о любом изменении заказа в кратчайшее время
6. Обеспечивать прямой канал связи с контролирующим диспетчером

### 2.1.2.2Схема взаимодействия классов и структурных объектов программы



Рисунок 6. Схема классов клиента

Список объектов:

1. Main.java – основной класс приложения, контролирующий работу остальных классов приложения.
2. FeedbackWork.java – класс, обеспечивающий обратную связь водителя с диспетчером, через текстовые сообщения, либо через звонок
3. GlobalVariables.java – класс, хранящий в себе очень редко меняющиеся переменные, используемые во всем приложении и сильно влияющие на ход его выполнения.
4. SQLWork.java - класс, обеспечивающий удаленную работу приложения с БД на сервере. Выполняет входящую команду и возвращает результирующий набор
5. ConnectionWork.java – класс, обеспечивающий соединение программы с интернетом, для получения данных через мобильные сети.
6. Internal Database - внутренняя база данных, хранящая в себе резервную копию самой необходимой водителю информации. Предназначена для ситуаций с отсутствием сигнала мобильных сетей, что не является большой редкостью даже в крупных городах
7. External Database – внешняя база данных (база данных сервера), хранящая в себе основную информацию, используемую водителем для доставки.
8. Website – веб-сайт с встроенным функционалом Google Maps для определения положения автомобиля и визуализации запланированного маршрута на карте с указанием возможных препятствий

# 2.2 Схема базы данных

База данных сервера должна будет хранить всю необходимую для работы системы информацию, при этом обеспечивая её целостность и непротиворечивость. Структура базы максимально упрощена (насколько возможно) и представлена на рисунке 7.

База данных рассмотренная мной в данной работе является не полной, но лишь минимально необходимой для полноценной работы системы. В частности, тут не присутствует возможность хранения информации о местоположении курьеров в разные моменты времени и нет возможности использовать эту базу в качестве хранилища паролей для защиты приложения от неавторизованного доступа

## 2.2.1 Список таблиц базы:

1. Table\_Orders
2. Table\_Customers
3. Table\_Meals
4. Table\_Recipes
5. Table\_Measurments
6. Table\_Ingredients
7. Table\_Batches
8. Table\_Locations
9. Table\_Suppliers
10. Table\_Status
11. Table\_MealsList
12. Table\_Cars
13. Table\_Overseers
14. Table\_SuppliersIngredientsList
15. Table\_Drivers

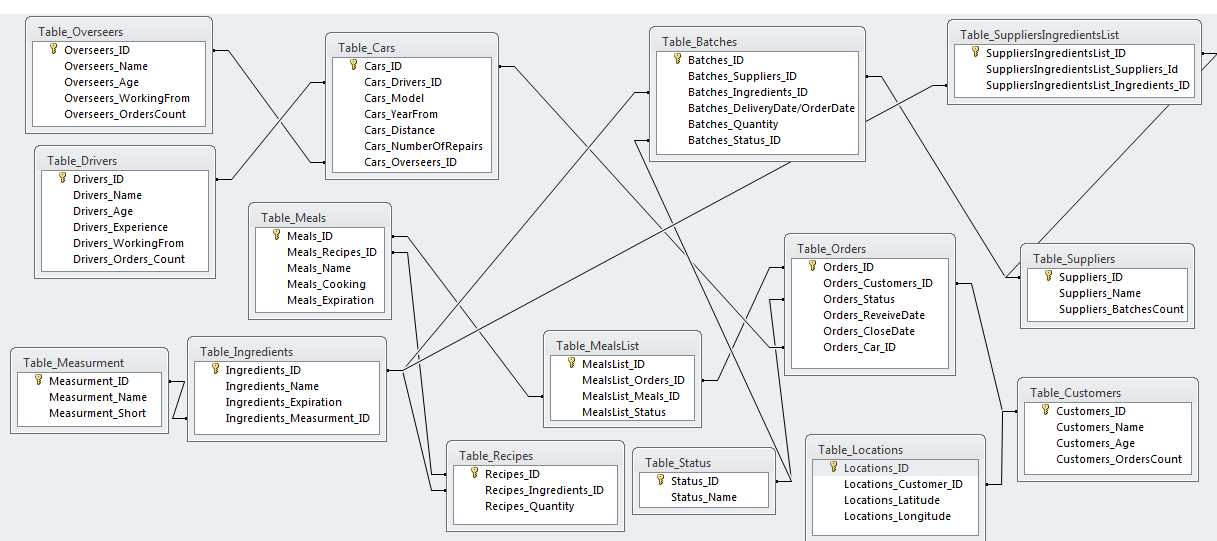


Рисунок 7 Схема данных БД

## 2.2.2 Описание структуры данных

### 2.2.2.1 Table\_Orders:

Эта таблица хранит в себе данные о заказах, сделанных клиентами, как об активных, так и о выполненных, в целях сбора статистики и построения модели потребности ресторана в ингредиентах.

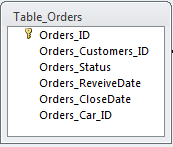


Рисунок 8 Таблица заказов

#### Поля таблицы:

1. Orders\_ID – (integer)уникальный идентификатор заказа в системе, также является первичным ключом. Уникальное поле, которое не может повторяться. Является индексом
2. Orders\_Customer\_ID – (integer)уникальный идентификатор пользователя, сделавшего заказ. Является внешним ключом (таблица: Table\_Customers)
3. Orders\_Status – (boolean)поле, которое показывает статус заказа, принимает значения (0 – заказ не выполнен и 1 – заказ закрыт)
4. Orders\_ReceiveDate – (datetime) поле, которое хранит дату и время поступления заказа в систему
5. Orders\_CloseDate – (datetime) поле, хранящее информацию о дате и времени закрытия заказа
6. Orders\_Car\_ID – (integer) внешний ключ таблицы Table\_Car, указывающий машину, на которую был назначен (которая выполнила) заказ.

### 2.2.2.2 Table\_MealsList:

В каждом заказе может быть несколько блюд, поэтому для их хранения в рамках одной записи заказа создаем отдельную таблицу.



Рисунок 9. Таблица списка блюд в заказе

#### Поля таблицы:

1. MealsList\_ID – (integer) уникальный идентификатор записи. Не может повторяться и является первичным ключом таблицы
2. MealsList\_Orders\_ID – (integer) внешний ключ из таблицы Table\_Orders. Показывает к какому заказу относятся блюда из списка
3. MealsList\_Meals\_ID – (integer) внешний ключ из таблицы Table\_Meals, показывающий идентификатор блюда, входящего в заказ
4. MealsList\_Status – (boolean) поле хранящее статус готовности блюда (0 – не готово, 1 – готово и можно отправлять)

### 2.2.2.3 Table\_Customers:

Таблица содержит данные о заказчиках для сбора статистики, поиска постоянных клиентов и изучения клиентов и их предпочтений.

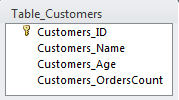


Рисунок 10. Таблица клиентов

#### Поля таблицы :

1. Customers\_ID – (integer) уникальный идентификатор покупателя, являющийся первичным ключом и индексом таблицы
2. Customers\_Name – (varchar) ФИО покупателя
3. Customers\_Age – (integer) возраст покупателя, используется для составления статистики предподчтений
4. Customers\_OrdersCount – (integer) количество заказов, сделанных покупателем. Собирается для статистики.

### 2.2.2.4 Table\_Meals:

В этой таблице хранится информация о блюдах, которые ресторан готовит и доставляем своим клиентам.



Рисунок 11. Таблица блюд

#### Поля таблицы:

1. Meals\_ID – (integer) уникальный идентификатор записи. Является первичным ключом таблицы, а так же индексом для ускорения поиска блюда под ID
2. Meals\_Recipes\_ID – (integer) внешний ключ таблицы Table\_Recipes, который устанавливает связь между блюдом и его рецептом
3. Meals\_Name – (varchar) название блюда
4. Meals\_Cooking – (memo) поле, хранящее в себе последовательность действий при приготовлении блюда
5. Meals\_Expiration – (integer) срок годности блюда после его приготовления

### 2.2.2.5 Table\_Recipes:

Эта таблица хранит в себе все ингредиенты с помощью которых готовится блюдо, а так же необходимое для приготовления количество каждого ингредиента в разных единицах измерений

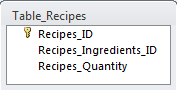


Рисунок 12. Таблица рецептов

#### Поля таблицы:

1. Recipes\_ID – (integer) является уникальным идентификатором каждой строки. Так же является индексом
2. Recipes\_Ingredients\_ID – (integer) внешний ключ, являющийся первичным ключом таблицы Table\_Ingredients. Используется для привязки ID каждого ингредиента к рецепту
3. Recipes\_Quantity – (float) поле, описывающее необходимое количество ингредиента для приготовления блюда

### 2.2.2.6 Table\_Measurment

Таблица хранит в себе данные о единицах измерения, используемых в приложении.

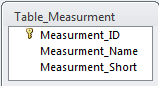


Рисунок 13. Таблица мер измерения

#### Поля таблицы:

1. Measurment\_ID – (integer) поле, являющееся первичным ключом таблицы, а так же её индексом.
2. Measurment\_Name – (varchar) название единицы измерения, применяемое в программе
3. Measurment\_Short – (varchar) – сокращение названия единицы измерения, для применения вместе с цифрами

### 2.2.2.7 Table\_Ingredients

Таблица содержит данные об ингредиентах, их принятом в системе названии и сроке годности.

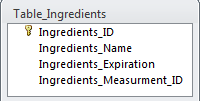


Рисунок 14. Таблица ингредиентов

#### Поля таблицы:

1. Ingredients\_ID – (integer) уникальный идентификатор рецепта, так же являющийся первичным ключом и индексом для ускорения поиска по ингредиентам
2. Ingredients\_Name – (varchar) название ингредиента, используемое информационной системой.
3. Ingredients\_Expiration – (integer) число дней, в течение которых ингредиент можно использовать
4. Ingredients\_Measurment\_ID – (integer) внешний ключ, привязывающий к каждому ингредиенту подходящие меры измерения. Является внешним ключом

### 2.2.2.8 Table\_Batches

Таблицы хранит в себе данные о партиях ингредиентов, поставленных поставщиками, их сроке годности и наличии на складе.

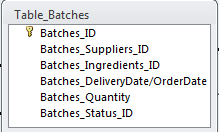


Рисунок 15. Таблица заказов на ингредиенты

#### Поля таблицы:

1. Batches\_ID – (integer) уникальный идентификатор поставки, являющийся первичным ключом таблицы, а так же индексом, созданным для ускорения поиска партии по ID
2. Batches\_Suppliers\_ID – (integer) внешний ключ, связывающий Table\_Batches с Table\_Suppliers. Указывает ID поставщика, поставившего компании партию ингредиентов.
3. Batches\_Ingredients\_ID – (integer) внешний ключ, связывающий таблицы по ID ингредиента, определяет тип ингредиента, содержащийся в данной поставке
4. Batches\_DeliveryDate/OrderDate – (datetime) время и дата доставки данной партии ингредиентов на склад или заказа партии, через систему у поставщика.
5. Batches\_Quantity – (integer) количество ингредиентов, поставленных в рамках данной партии
6. Batches\_Status\_ID – (integer) внешний ключ, устанавливающий связь между таблицами Table\_Batches и Table\_Status. Указывает статус партии в системе (0 – заказана, но не доставлена, 1 – доставлена на склад, 2- партия закончилась, 3- у партии истек срок годности)

### 2.2.2.9 Table\_Locations:

Таблица содержит данные о местонахождении клиентов, для более быстрой доставки блюд по адресу постоянным покупателям.



Рисунок 16. Таблица местоположений клиентов

#### Поля таблицы:

1. Locations\_ID – (integer) уникальный идентификатор строки. Является также первичным ключом таблицы.
2. Locations\_Customer\_ID – (integer) внешний ключ, связывающий Table\_Locations с Table\_Customers. Предназначен для создания связи между покупателем и его домашним адресом
3. Locations\_Latitude – (integer) широта точки местонахождения клиента, для визуализации через Google Maps.
4. Locations\_Longitude – (integer) долгота точки местонахождения клиента, для визуализации через Google Maps.

### 2.2.2.10 Table\_Suppliers:

Таблица предназначена для хранения информации о поставщиках ингредиентов для ресторана.



Рисунок 17. Таблица поставщиков

#### Поля таблицы:

1. Suppliers\_ID – (integer) уникальный идентификатор каждого поставщика в системе. Является первичным ключом, а так же индексом.
2. Suppliers\_Name – (varchar) название поставщика, используемое внутри системы.
3. Suppliers\_BatchesCount – (integer) количество заказов на ингредиенты, выполненных поставщиком

### 2.2.2.11 Table\_Status:

Таблица хранит в себе данные о статусе партий ингредиентов, внесенных в систему.

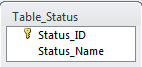


Рисунок 18. Таблица статусов заказов

#### Поля таблицы:

1. Status\_ID – (integer) уникальный идентификатор статуса, являющийся так же первичным ключом.
2. Status\_Name – (varchar) название статуса, принятое в рамках данной системы.

### 2.2.2.12 Table\_SuppliersIngredientsList:

Таблица содержит в себе данные о списке ингредиентов, которые можно заказать у каждого поставщика

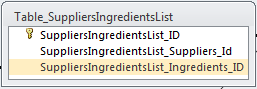


Рисунок 19. Таблица ингредиентов, поставляемых разными поставщиками

#### Поля таблицы:

1. SuppliersIngredientsList\_ID – (integer) уникальный идентификатор каждой записи в таблице. Является первичным ключом в таблице и индексом.
2. SuppliersIngredientsList\_Suppliers\_ID – (integer) внешний ключ, связанный с таблицей Table\_Suppliers, показывает какой поставщик поставляет данный ингредиент
3. SuppliersIngredientList\_Ingredients\_ID – (integer) внешний ключ, связывающий данную таблицу с таблицей Table\_Ingredients, указывает какой ингредиент поставляет данный поставщик

### 2.2.2.13 Table\_Cars:

Таблица содержит в себе данные о машинах, используемых компанией для доставки.

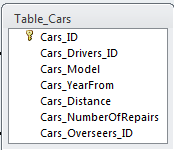


Рисунок 20. Таблица машин компании

#### Поля таблицы:

1. Cars\_ID – (integer) уникальный идентификатор машины, используемый в системе. Является первичным ключом и индексом.
2. Cars\_Drivers\_ID – (integer) ID водителя данной машины. Является внешним ключом, связывающим данную таблицу с таблицей Table\_Drivers.
3. Cars\_Model – (varchar) модель машины
4. Cars\_YearFrom – (datetime) год покупки машины
5. Cars\_Distance – (integer) пробег машины за время работы в компании по доставке.
6. Cars\_NumberOfRepairs - (integer) число ремонтов данной машины
7. Cars\_Overseers\_ID – (integer) каждая машина привязана к определенному диспетчеру. Поле является внешним ключом для связи с таблицей Table\_Overseers

### 

### 2.2.2.14 Table\_Overseers:

Таблица будет содержать в себе данные о диспетчерах, контролирующих поток заказов и назначающих их на машины.

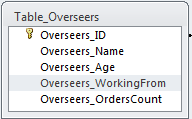


Рисунок 21. Таблица диспетчеров

#### Поля таблицы:

1. Overseers\_ID – (integer) уникальный идентификатор диспетчера в системе. Является первичным ключом и индексом БД
2. Overseers\_Name – (varchar) ФИО диспетчера
3. Overseer\_Age – (integer) возраст диспетчера
4. Overseers\_WorkingFrom – (datetime) дата приема диспетчера на работу
5. Overseers\_OrdersCount – (integer) число заказов, обработанных данным диспетчером

### 2.2.2.15 Table\_Drivers:

Таблица содержит в себе данные о водителях, нанятых на работу в ресторан.

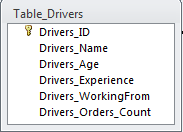


Рисунок 22. Таблица водителей

#### Поля таблицы:

1. Drivers\_ID – (integer) уникальный идентификатор водителя в системе. Является первичным ключом.
2. Drivers\_Name – (varchar) ФИО водителя
3. Drivers\_Age – (integer) возраст водителя
4. Drivers\_Experience – (integer) водительский стаж
5. Drivers\_WorkingFrom – (datetime) дата приема водителя на работу
6. Drivers\_OrdersCount – (integer) количество заказов, выполненных водителем

## 2.2.3 Использование структуры данных в ИС

Приведенная выше структура данных хранит в себе всю необходимую информацию для корректной работы системы по всем аспектам доставки. Система предназначена для постоянного обновления списков заказов и поставок (Tables\_Batches, Table\_Orders), на основе которых и выполняется главная часть обоих БП. Остальные таблицы хранят вспомогательную информацию (имена диспетчеров и водителей, рецепты), которая имеет не меньшую важность, но обновляется намного реже двух главных таблиц.

В основном система будет работать именно с Table\_Orders и Table\_Batches, т.к. они хранят в себе большую часть необходимой для выполнения заказа информации, остальные будут использоваться лишь при необходимости. Это означает, что большая часть функционала будет оптимизирована для работы с двумя главными таблицами.

# 3.1 Руководство пользователя

В данной работе я приведу часть описания выполнения одного из наиболее важных процессов ресторана – получения и назначения заказов на машину. Всего в приложении сервера задействовано 8 форм, как показано на рисунке 23.

## 3.1.2 Схема взаимодействия форм в приложении



Рисунок . Схема взаимодействия форм в приложении

В рассматриваемом процессе принимают участие 4 формы в обязательном порядке и еще одна в случае необходимости добавления нового клиента в базу данных.

## 3.1.3 Этапы процесса

Процесс состоит из трех главных этапов:

3.1.3.1 Аутентификация и вход в систему

Окно формы аутентификации приводится на рисунке № 24. Для входа в системы необходимо выбрать в верхнем выпадающем меню систему в которую вы хотите войти (в сервер входят две независимые системы и диспетчер обработки заказов клиентов не обязательно может иметь доступ к системе поставок). После выбора системы пользователь вводит комбинацию логина и пароля и в случае правильной комбинации после проверки получает доступ к функционалу выбранной системы.

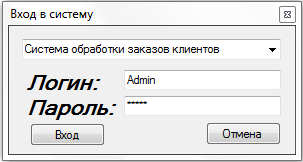


Рисунок Форма аутентификации

3.1.3.2 Ввод заказа в базу и ожидание его готовности

После прохождения процедуры аутентификации диспетчер попадает на вкладку «Прием заказов клиентов» основной формы подсистемы, в которой можно добавлять новые заказы и редактировать или удалять предыдущие (изображена на рисунке № 25). Диспетчер может найти необходимую ему запись по ключевым словам, либо отфильтровать таблицу по нужному столбцу. Таблица содержит данные о списке блюд в заказе, дате совершения заказа, степени готовности блюд и о заказчике (что в том числе дает привязку заказа к местоположению).

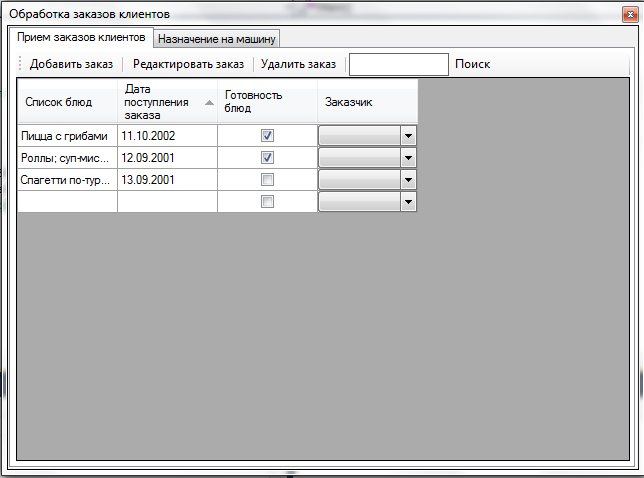


Рисунок Форма для обработки заказов клиентов

Добавление нового заказа или редактирование старого происходит через отдельную, специально предназначенную для этой операции форму (рисунок 26). В этой форме диспетчер может ввести клиента, список блюд в заказе и дату\время совершения заказа, а после – сохранить изменения.

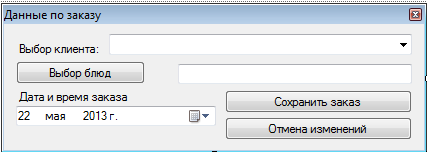


Рисунок Форма данных о заказе

При нажатии на кнопку «Выбор блюд» в форме добавления данных по заказу происходит открытие отдельной формы, в которой отображается список блюд, доступных для заказа и диспетчер по запросу пользователя выбирает необходимые путем проставления меток в логическом поле таблицы. С помощью верхней панели интерфейса диспетчер может запустить поиск по названию блюда, либо сохранить\отменить изменения в только что созданном наборе блюд.

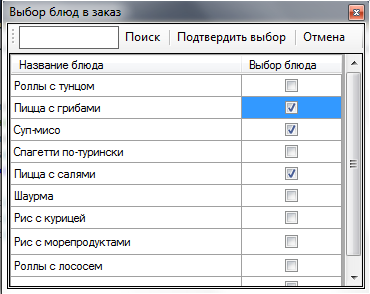


Рисунок Форма выбора блюд в заказ

При необходимости, путем выбора в верхнем выпадающем меню формы «Данные по заказу» значения «Добавить нового клиента…» можно открыть форму добавления клиента (изображена на рисунку № 28), в которой диспетчер имеет возможность задать ФИО клиента, а так же набор адресов, по которому будет производиться доставка блюд покупателю. Этот набор необходимо редактировать, путем проставления меток в логическом поле «Доставка по адресу», чтобы одновременно был отмечен только один адрес, необходимый для доставки. В верхнем меню формы находятся кнопки, используемые для подтверждения изменений, либо отмены данного изменения.

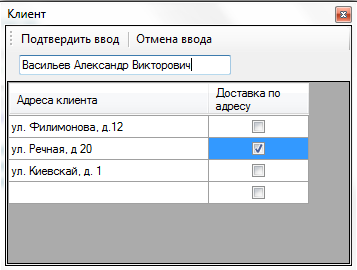


Рисунок Форма данных о клиенте

### 3.1.3.3 Назначение заказа на машину

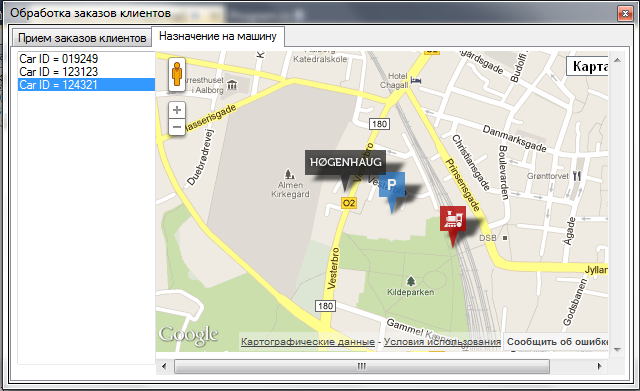
После того, как заказ выполнен поварами, он будет отображаться на карте (расположена в основной форме на вкладке «Назначение на машину», рисунок 29) в виде маркера, который необходимо перетащить на идентификатор машины для назначения заказа на неё. На карте так же отображаются маршруты транспортного средства, препятствия (перекрытые дороги, пробки). В случае выполнения заказа в систему приходит сигнал с клиентского приложения на Android и заказ исчезает из списка назначенных на машину. На этом процесс обработки заказа заканчивается.

Рисунок Назначение на машину

# Заключение

Целью работы было – улучшение существующих бизнес-процессов логистики с целью ускорения и упрощения их выполнения путем унификации всех данных и функций работы с ними в рамках одной системы, доступной как диспетчерам и водителям внутри компании, так и внешним поставщикам, клиентам, защищая конфиденциальную информацию с помощью системы разграничения доступа.

По сравнению с состоянием «до внедрения системы» данное программное решение являет огромным шагом вперед в улучшении качества БП. Программа ускоряет множество операций, требовавших немало времени для исполнения и открывает перед компанией новые возможности.

В процессе написания данной работы было:

1. Спроектирована структура программного решения, на базе которого возможно провести полноценную разработку ИС
2. Была создана структура базы данных для хранения информации, используемой приложением.
3. Разработана логика программного решения и обоснована актуальность его применения в данной области.
4. Рассмотрены альтернативные варианты программных решений.

Данный проект не закончен и является лишь схемой приложения, но которой можно создать намного более централизованную и удобную систему. Сейчас сервер является десктопным приложением с удаленным доступом, но в идеале его стоит объединить с сайтом, через который клиенты оставляют заказы, а так же наладить полноценную систему для обратной связи с клиентами в рамках приложения. В будущем для развития приложения необходимо усовершенствовать систему запросов ингредиентов, добавив туда полноценную систему контроля над складом и программное решение, способное на основе статистических данных предсказывать спрос клиентов на некий отрезок времени в будущем для предварительного обеспечения ингредиентами.

В целом система уже является качественным шагом вперед по сравнению с предыдущим состоянием БП компании, но еще один её плюс, в том, что модульное строение системы открывает просторы для дальнейшего её улучшения в рамках следующих проектов, что теоретически позволит ей прослужить сравнительно долгий срок.

# Список литературы:

1. ru.wikipedia.org/wiki/1С:Предприятие
2. ru.wikipedia.org/wiki/ERP
3. <http://solutions.1c.ru/catalog/gps/features>
4. <http://solutions.1c.ru/catalog/tms/features>
5. Теория и практика построения баз данных. /Д. Крёнке – М.: «Питер», 2005
6. C# и платформа .Net / Эндрю Троелсен, 2004
7. Programming the Mobile Web, 2nd Edition  
   Reaching Users on iPhone, Android, BlackBerry, Windows Phone, and more. / Maximiliano Firtman, 2013
8. Android Application Development Cookbook: 93 Recipes for Building Winning Apps/ Wei-Meng Lee, 2013
9. Базы данных. Проектирование, реализация и сопровождение. Теория и практика./ Томас Коннолли, Каролин Бегг, 2003
10. Системы баз данных. Полный курс./ Гектор Гарсиа-Молина, Джеффри Ульман, Дженнифер Уидом, 2006

# Список сокращений:

1. Бизнес-процесс - БП
2. Программное обеспечение – ПО
3. Информационная система - ИС