Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет  
Кафедра экономической информатики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКАк курсовой работе по курсу:  
«Средства и технологии анализа и разработки информационных систем»

на тему:

«РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ УЧЕТА И РЕГИСТРАЦИИ ЗАКАЗОВ КЛИЕНТОВ НА ПРОВЕДЕНИЕ БАНКЕТОВ»

Студенток Шульи Полины Олеговны

группы 773601 Лещинской Дарьи Сергеевны

Руководитель Сторожев Дмитрий Алексеевич

Минск 2020

**Содержание**

[Введение 5](#_Toc34744382)

[1 Описание учета и регистрации заказов клиентов на проведение банкетов 6](#_Toc34744383)

[1.1 Описание процесса оформления заказа на банкет 6](#_Toc34744384)

[1.2 Сравнительная характеристика системы оформления банкетов в реальном мире и web-приложения 8](#_Toc34744385)

[2 Функциональное моделирование 10](#_Toc34744386)

[2.1 Описание системы в виде «Как есть» 11](#_Toc34744387)

[2.2 Описание системы в виде «Как должно быть» 15](#_Toc34744388)

[3 Спецификация вариантов использования системы 22](#_Toc34744389)

[4 Информационная модель системы 24](#_Toc34744390)

[5 Обоснование выбора компонентов и технологий 27](#_Toc34744391)

[5.1 Диаграмма последовательности 27](#_Toc34744392)

[5.2 Диаграмма компонентов 29](#_Toc34744393)

[5.3 Диаграмма развертывания 30](#_Toc34744394)

[5.4 Обоснование выбора технологий 30](#_Toc34744395)

[6 Модели представления системы и их описание 33](#_Toc34744396)

[7 Руководство пользователя 34](#_Toc34744397)

[8 Результаты тестирования разработанной системы 35](#_Toc34744398)

[Заключение 36](#_Toc34744399)

[Список используемых источников 37](#_Toc34744400)

[Приложение А Листинг программы 38](#_Toc34744401)

# Введение

Очень важно успевать за развитием мира. Появление новых технологий заставляет нас изучать и использовать их для своих нужд.

При оформлении услуг по организации банкета в офисе мы тратим огромное количество времени на поиск нужного адреса, оформление бумаг, оплату и тому подобное. При этом мы не может иметь стопроцентную уверенность, что попадем в рабочие часы, что наш заказ оформят без очереди или что наши личные данные будут правильно внесены в договор или чек об оплате. Однако в современном мире появилась возможность оформления услуг с помощью веб-приложения, где заказ может быть отправлен на регистрацию в любое время дня и ночи, а также без очередей, с возможностью оплатить услуги онлайн и собственноручно проверить введенные данные.

Разработанное программное средство сможет быстро, точно, просто оформить заказ на банкет, а также произвести отчёт о проделанной работе.

Цель написания курсового проекта состоит в автоматизации процесса оформления и учета банкетов, а также в сокращении числа ошибок при работе с объемными данными, которые используются при их регистрации.

Для достижения данной цели необходимо решить следующие задачи:

− описание бизнес-процесса работы компании по предоставлению услуг банкетов (IDEF0);

− функциональное моделирование процессов предметной области;

− разработка алгоритмов работы приложения;

− создание абстрактной системы работы приложения при помощи UML-диаграмм;

− разработка клиент-серверного взаимодействия;

− разработка интерфейса программы для удобной и эффективной работы онлайн-сервиса по регистрации и учету заказов клиентов на проведение банкетов;

− тестирование программного продукта.

# ОПИСАНИЕ **УЧЕТА И РЕГИСТРАЦИИ ЗАКАЗОВ КЛИЕНТОВ НА ПРОВЕДЕНИЕ БАНКЕТОВ**

## Описание процесса оформления заказа на банкет

Ни одно торжественное мероприятие, будь то день рождения, юбилей, свадьба или корпоратив не обходится без организации банкета.

Правильная организация приема заказа на банкет способствует успешному его проведению.

Заказ на обслуживание банкета принимает менеджер или метрдотель. Он уточняет:

* вид банкета и предлагает заказчику ознакомиться с помещением;
* план расстановки столов и размещения гостей;
* согласовы­вает дату и время проведения банкета;
* количество участни­ков;
* состав по возрасту, полу, национальности;
* метод об­служивания (банкет с полным или частичным обслуживани­ем);
* предполагаемую стоимость заказа;
* особенности оформления банкетного зала;
* подачи аперитива;
* вид музы­кального обслуживания.

Заказчику можно предложить несколько вариантов за­ранее подготовленного меню или вновь составить его с уче­том возможного изготовления блюд и закусок, не входящих в меню ресторана. По желанию заказчика может быть при­нят заказ на фирменные кулинарные и кондитерские изде­лия, изготовляемые предприятием. Стоимость заказа опла­чивается заказчиком по ценам калькуляций.

В заказе указывается номер телефона, адрес и фамилия заказчика, порядок расчета (наличный, безналичный или по кредитной карте). После предварительного согласования меню заказчик оплачивает 50% стоимости заказа, на при­нятые деньги ему выписывают приходный кассовый ордер и квитанцию к нему, которая вручается заказчику. Заказ регистрируется в специальной книге учета заказов на об­служивание торжеств. Одновременно работник, принимаю­щий заказ, знакомит заказчика с правилами работы пред­приятия, порядком обслуживания, возмещения возможных убытков по вине заказчика и гостей, с расценками на оплату аренды залов, составляет контракт.

При окончательном согласовании меню (не позднее чем за два дня до начала торжества) менеджер или метрдотель совместно с заказчиком составляет заказ-счет в пяти экзем­плярах, который утверждается директором предприятия и служит расчетным документом. Номер заказа-счета должен соответствовать номеру данного заказа в книге учета заказов Кассир, получив утвержденный заказ-счет и выписанный бухгалтерский приходный кассовый ордер, принимает от заказчика доплату за заказ, проставляет на всех экземплярах штамп «Оплачено», заверяет каждый экземпляр подписью. Первый экземпляр менеджер или метрдотель передает заказчику. Второй экземпляр заказа-счета остается у кассира до дня обслуживания. В день обслуживания он передает его в бухгалтерию вместе с кассовым отчетом; третий — официанту, которому поручено выполнение заказа; четвертый и пятый — на кухню и в сервис-бар. Четвертый и пятый экземпляры заказа-счета после выполнения заказа сдаются в бухгалтерию.

Продукцию из кухни и сервис-бара отпускают по чекам, печатаемым на принтерах согласно заказу-счету. В процессе обслуживания заказчик может сделать дополнительный заказ, который он оплачивает наличными деньгами. Дополнительный заказ фиксируется на обратной стороне заказа-счета. Об изменениях в заказе-счете заказчик должен предупредить не позднее чем за сутки до начала банкета. В этом случае заказчику могут быть возвращены деньги или предложены дополнительные услуги (подача шампанского, фруктов, коктейлей, аперитива, если это не было предусмотрено в заказе). Если отказ поступил в день проведения банкета, заказчик должен оплатить предприятию фактически понесенные расходы, т. е. стоимость блюд по продажным ценам, которые не могут быть реализованы (истек срок реализации, отсутствует возможность передать изделия для продажи в зале и др.). Заказчик может распорядиться с указанной продукцией по своему усмотрению.

По окончании проведения банкета для возмещения причиненного ущерба (бой или пропажа посуды, приборов, салфеток и нанесение других убытков предприятию по вине гостей) заказчику выписывают счет, а деньги приходуются кассой [1].

Заказчик имеет право отказаться от всего заказа не позднее чем за сутки до момента обслуживания.

Если не все участники смогут явиться на торжество, то заказчику по его письменному заявлению, поданному не позднее чем за четыре часа до начала мероприятия, могут быть отданы деньги в размере стоимости заказа.

Возврат денег производится на основании письменного заявления по расходному кассовому ордеру. Однако в этом случае заказчик обязан возместить ресторану стоимость приготовленной продукции [2].

Так же при планировании банкета присутствует возможность регистрации и оформления предварительного заказа.

Работник продавца не позднее чем за один календарный день до начала оказания услуг, связанных с осуществлением общественного питания, совместно с покупателем оформляет предварительный заказ по форме, установленной продавцом, в двух экземплярах, один из которых выдается покупателю.

Предварительный заказ должен содержать следующую информацию:

* наименование (фирменное наименование) и местонахождение продавца;
* фамилия, собственное имя, отчество и место жительства продавца (для индивидуального предпринимателя);
* фамилия, собственное имя, отчество покупателя;
* вид услуги, связанной с осуществлением общественного питания, ее особенности, предварительная цена;
* дата заключения предварительного заказа и дата его исполнения;
* другие сведения по усмотрению продавца и покупателя.

Изменения в предварительном заказе согласовываются с покупателем.

Покупатель вправе в любое время отказаться от предварительного заказа при условии оплаты продавцу фактически понесенных им расходов, связанных с отказом от исполнения предварительного заказа.

Предварительный заказ регистрируется в книге учета предварительных заказов, которая ведется продавцом в произвольной форме [3].

## Сравнительная характеристика системы оформления банкетов в реальном мире и web-приложения

Далее будет приведена сравнительная характеристика процессов в реальном мире и web-приложении по автоматизации учета и регистрации заказов банкетов (таблица 1.1).

Основными пунктами для сравнительной характеристики стали «Сокращение работы кадров», «Сокращение временных затрат», «Сокращение бюджета», «Ориентированность на клиентскую базу», «Устойчивость системы» и «Круглосуточное обслуживание».

Автоматизированная система обладает однозначным преимуществом на фоне системы в реальном времени. При использовании веб-приложения сокращается количество работы с бумажными документами и время общения с клиентами, в следствие чего уменьшаются временные затраты, работа кадров и бюджет на обслуживание проектов. Веб-приложение имеет широкую ориентированность на клиентскую базу, т.к. пользоваться им может одновременно неограниченное количество человек без ожидания своей очереди, как это происходит в реальном мире. Имеется возможность оформления заказа в ночное время, с одобрением его в рабочие часы компании, что является основным плюсом для клиентов с ненормированным графиком работы. И конечным, главным плюсом автоматизированной системы будет то, что она имеет проверку на ввод и анализ данных, что является несомненным гарантом качества производимой системы.

Таблица 1.1 – Сравнительная характеристика процессов в реальном мире и автоматизированной системе

|  |  |  |
| --- | --- | --- |
| Название процесса | Система в реальном мире | Web-приложение |
| **1** | **2** | **3** |
| Сокращение работы кадров | - | + |
| Сокращение временных затрат | - | + |
| Сокращение бюджета | - | + |
| Ориентированность на клиентскую базу | - | + |
| Круглосуточное обслуживание | - | + |
| Устойчивость системы | - | + |

В следующей главе мы выделим основные задачи и рассмотрим методы их решения.

# Функциональное моделирование

Для наглядного примера система учета и регистрации банкетов будет показана с помощью нотации IDEF0 в двух видах: «Как есть» и «Как должно быть».

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. К ее особенностям можно отнести:

* использование контекстной диаграммы;
* поддержка декомпозиции;
* доминирование;
* выделение 4 типов стрелок.

Контекстная диаграмма. Самая верхняя диаграмма, на которой объект моделирования представлен единственным блоком с граничными стрелками. Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой.

Поддержка декомпозиции. Нотация IDEF0 поддерживает последовательную декомпозицию процесса до требуемого уровня детализации. Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский процесс, но описывает ее более подробно.

Доминирование. Блоки модели IDEF0 на неконтекстной диаграмме должны располагаться по диагонали - от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров. Блоки на диаграмме, расположенные вверху слева, "доминируют" над блоками, расположенными внизу справа. "Доминирование" понимается как влияние, которое блок оказывает на другие блоки диаграммы. Расположение блоков на листе диаграммы отражает авторское понимание доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные.

Выделение 4 типов стрелок. Выделяются следующие типы стрелок: "Вход", "Выход", "Механизм", "Управление". Входы преобразуются или расходуются процессом, чтобы создать то, что появится на его выходе. Управления определяют условия, необходимые процессу, чтобы произвести правильный выход. Выходы - данные или материальные объекты, произведенные процессом. Механизмы идентифицируют средства, поддерживающие выполнение процесса. Таким образом, блок IDEF0 показывает преобразование входа в выход с помощью механизмов с учетом управляющих воздействий [4].

Далее будет показана система в виде «Как есть».

## 2.1 Описание системы в виде «Как есть»

Описание системы в виде «Как есть» представляет собой учет и регистрацию заказов на проведение банкетов в реальной жизни.

На рисунке 2.1 показана контекстная диаграмма «Организовать учёт и регистрацию банкетов». «Управлением» в данной системе являются «Трудовой кодекс», «Шаблоны заявлений», «Законодательство РБ»; «Механизмами» являются «Организатор», «Кассир», «Консультант», «Бухгалтер», «Клиент»; «Входом» являются «Информация о банкете», «Паспортные данные», «Скидочная карта»; «Выходом» являются «Договор», «Чек об оплате» и «Отчёт по банкетам».

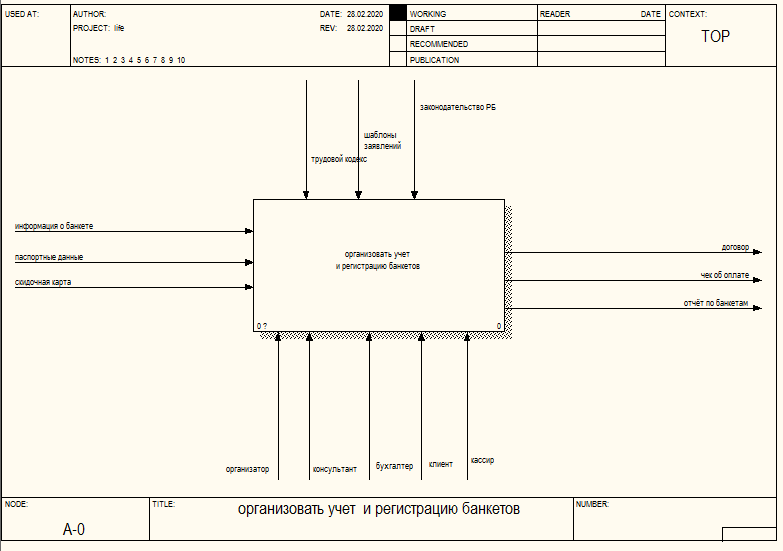


Рисунок 2.1 – Контекстная диаграмма системы «Как есть»

На рисунке 2.2 показана декомпозиция контекстной диаграммы. Она включает в себя блоки «Оформить договор в офисе», «Оплатить договор в банке», «Подготовить банкет» и «Произвести учёт проведенных банкетов». Оформление банкетов происходит следующим образом: сначала происходит консультация, обсуждаются требуемые условия и заключается договор; затем происходит оплата в ближайшем отделении банка; после подтверждения оплаты в офисе, компания начинает подготовку оформления банкета и после проведения рассчитывается с наемным персоналом; в конце бухгалтер имеет возможность составить отчёт о проведенных банкетах.

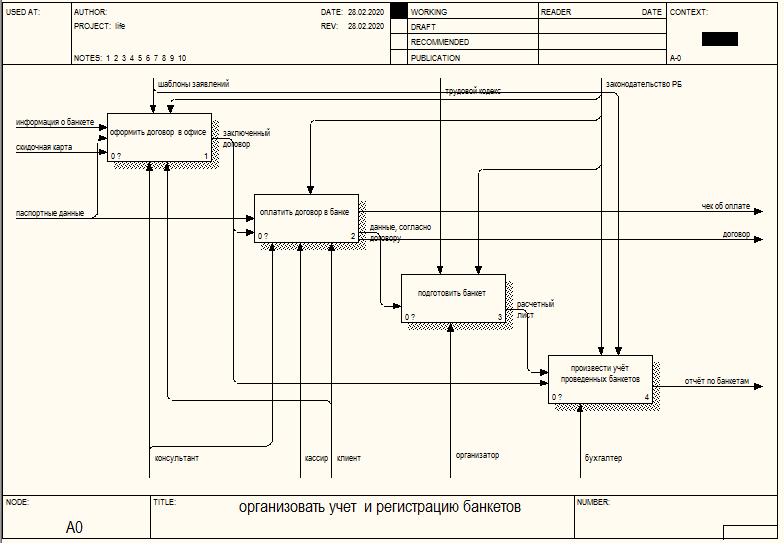


Рисунок 2.2 – Декомпозиция контекстной диаграммы

На рисунке 2.3 показана декомпозиция процесса «Оформить договор в офисе». Она включает в себя блоки «Выбрать условия банкета», «Распечатать договор», «Заполнить договор», «Предъявить скидочную карту» и «Заключить договор».

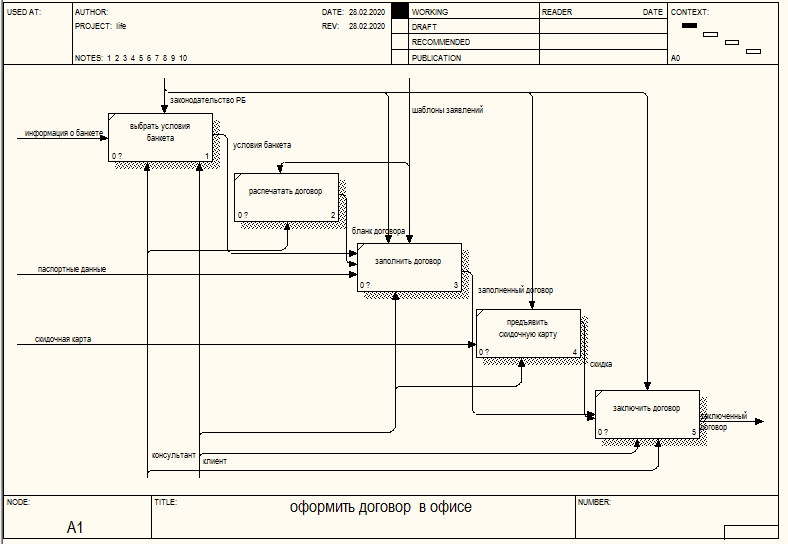


Рисунок 2.3 – Декомпозиция процесса «Оформить договор в офисе»

На рисунке 2.4 показана декомпозиция процесса «Оплатить договор в банке». Она включает в себя блоки «Предъявить данные реквизитов для оплаты», «Оплатить согласно договору в кассе», «Отнести чек в офис» и «Зарегистрировать договор».

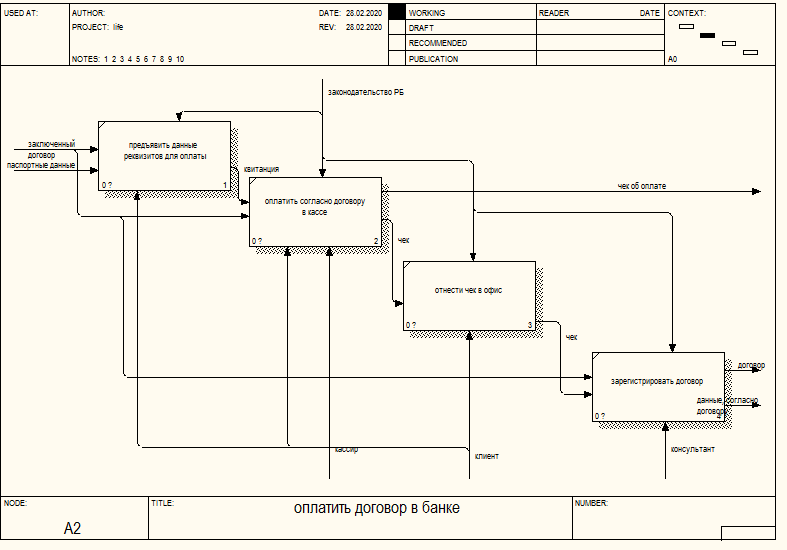


Рисунок 2.4 – Декомпозиция процесса «Оплатить договор в банке»

На рисунке 2.5 показана декомпозиция процесса «Подготовить банкет». Она включает в себя блоки «Забронировать локацию», «Забронировать работников» и «Оплатить работу наемного персонала и локацию».

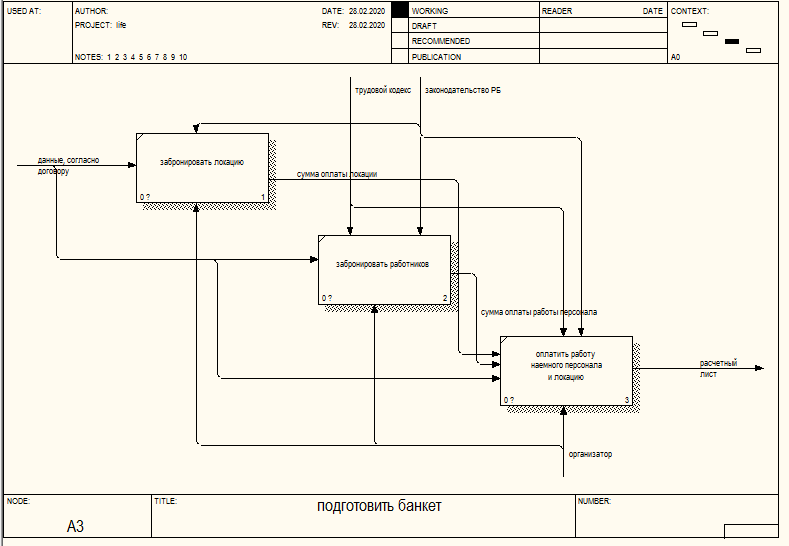


Рисунок 2.5 – Декомпозиция процесса «Подготовить банкет»

На рисунке 2.6 показана декомпозиция процесса «Произвести учёт проведенных банкетов». Она включает в себя блоки «Собрать статистику о проведенных банкетах» и «Составить отчёт».

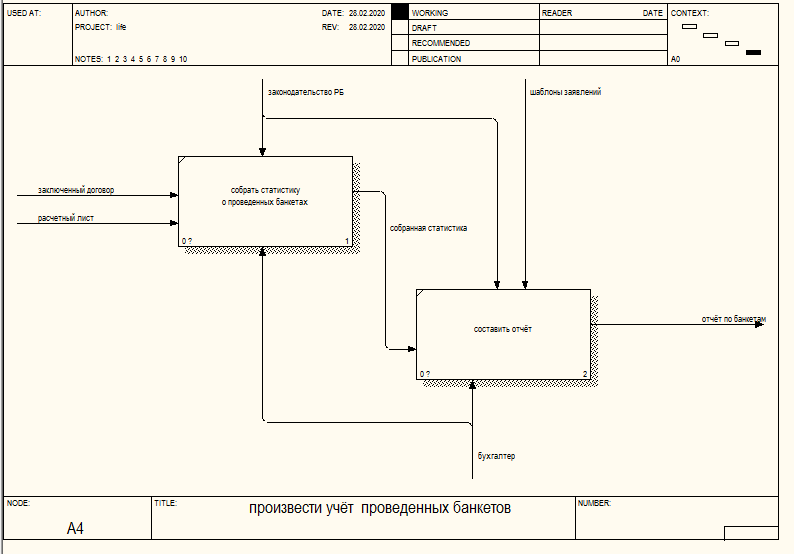


Рисунок 2.6 – Декомпозиция процесса «Произвести учет проведенных банкетов»

В следующем подразделе будет представлено описание системы «Как должно быть».

## 2.2 Описание системы в виде «Как должно быть»

Описание системы в виде «Как должно быть» представляет собой учет и регистрацию заказов на проведение банкетов в виде готового веб-приложения.

На рисунке 2.7 показана контекстная диаграмма «Автоматизировать систему учеба и регистрации заказов клиентов на проведение банкетов». «Управлением» в данной системе являются «Трудовой кодекс», «Система менеджмента качества», «Законодательство РБ»; «Механизмами» являются «Приложение», «Сотрудник», «Клиент»; «Входом» являются «Информация о заказе», «Информация о промокоде», «Паспортные данные», «Данные карты»; «Выходом» являются «Заключенный договор», «Чек об оплате», «Отчёт о проведенных банкетах» и «Сообщение о регистрации заказа».

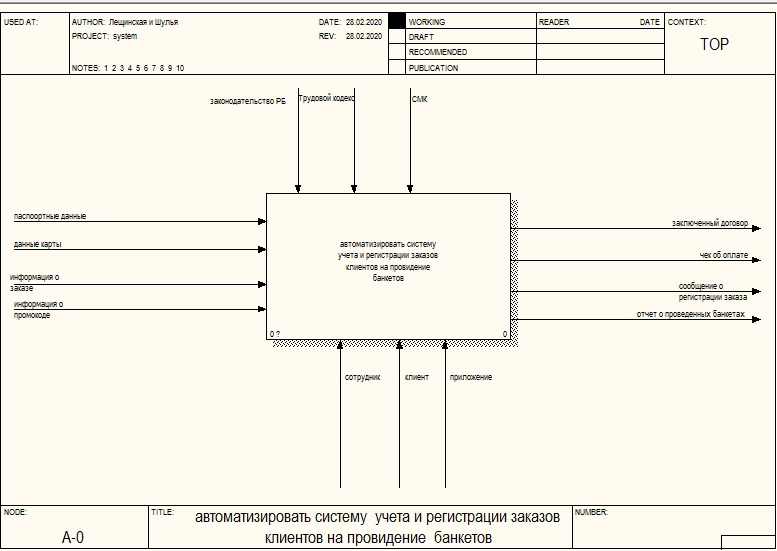


Рисунок 2.7 – Контекстная диаграмма системы «Как должно быть»

На рисунке 2.8 показана декомпозиция контекстной диаграммы. Она включает в себя блоки «Принять заказ», «Оформить заказ», «Обслужить банкет» и «Произвести учёт банкетов». Оформление банкетов в веб-приложении происходит следующим образом: в процессе оформления банкета клиент выбирает интересующие его позиции меню, развлечений и д.р; далее автоматически происходит формирование шаблона заявление в приложении с полями, куда клиент должен ввести паспортные и платёжные данные; данные договор отправляется в офис и происходит регистрация банкета; после его проведения сотрудники компании через приложение переводят деньги на счёт наёмных рабочих и локации банкета; при желании администратор может просмотреть в приложении отчёт по проведенным банкетам.

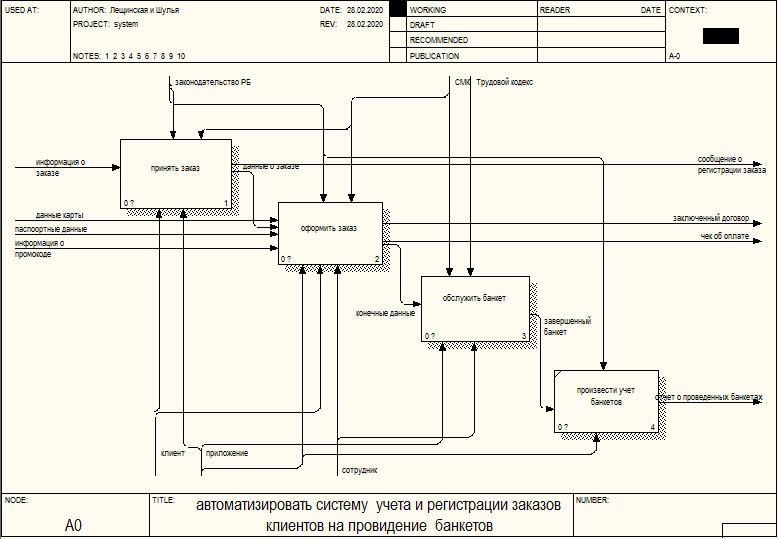


Рисунок 2.8 – Декомпозиция контекстной диаграммы

На рисунке 2.9 показана декомпозиция процесса «Принять заказ». Она включает в себя блоки «Выбрать место проведения», «Выбрать дату», «Выбрать меню», «Выбрать развлечение», «Выбрать количество человек», «Выбрать тематику банкета» и «Сохранить данные».

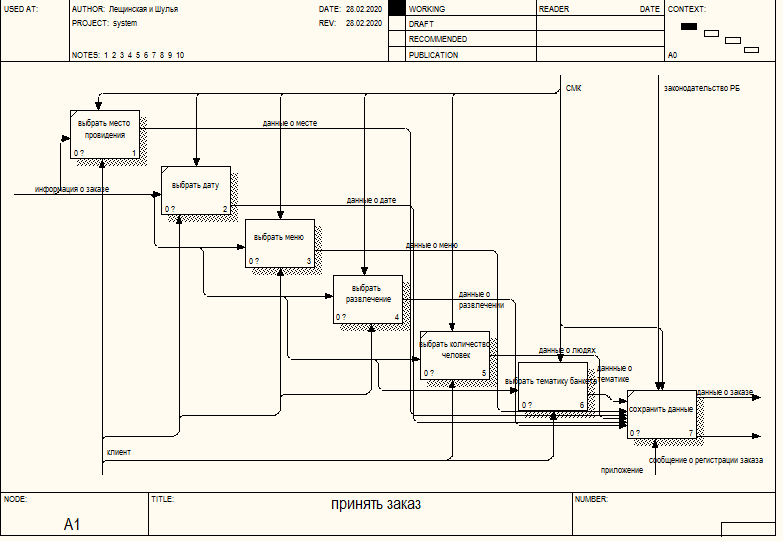


Рисунок 2.9 – Декомпозиция процесса «Принять заказ»

На рисунке 2.10 показана декомпозиция процесса «Выбрать меню». Она включает в себя блоки «Выбрать напитки», «Выбрать закуски», «Выбрать горячие блюда» и «Выбрать десерты».

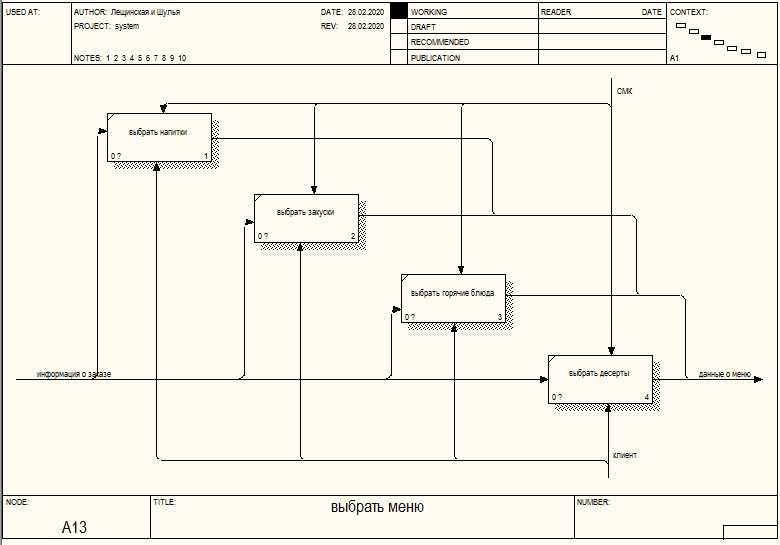


Рисунок 2.10 – Декомпозиция процесса «Выбрать меню»

На рисунке 2.11 показана декомпозиция процесса «Выбрать развлечение». Она включает в себя блоки «Заказать певца», «Заказать тамаду», «Заказать танцевальное шоу» и «Заказать видеосъёмку».

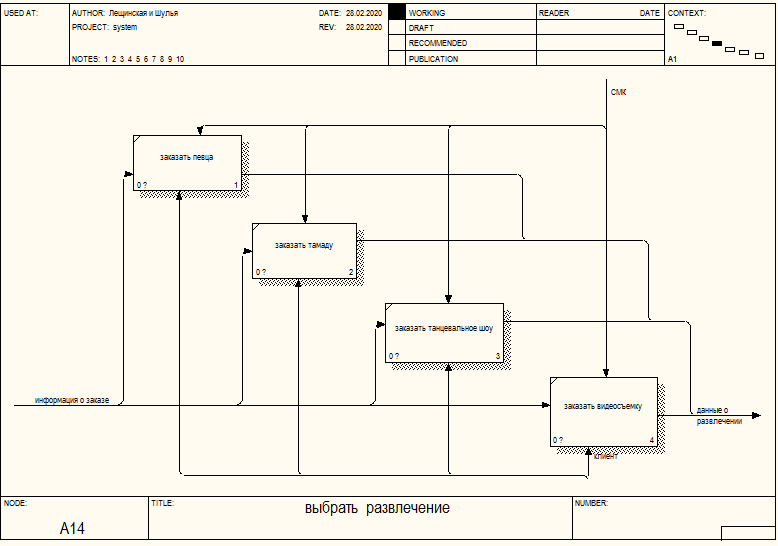


Рисунок 2.11 – Декомпозиция процесса «Выбрать развлечение»

На рисунке 2.12 показана декомпозиция процесса «Выбрать тематику банкета». Она включает в себя блоки «Выбрать банкет с полным обслуживанием», «Выбрать банкет с частичным обслуживанием», «Выбрать банкет-фуршет», «Выбрать банкет-чай» и «Выбрать дипломатический приём».

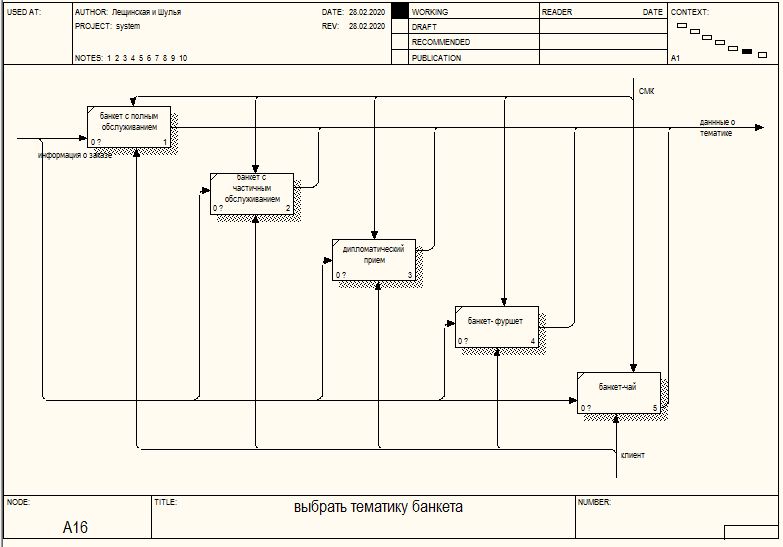


Рисунок 2.12 – Декомпозиция процесса «Выбрать тематику банкета»

На рисунке 2.13 показана декомпозиция процесса «Оформить заказ». Она включает в себя блоки «Заключить договор», «Ввести промокод» и «Оплатить заказ».

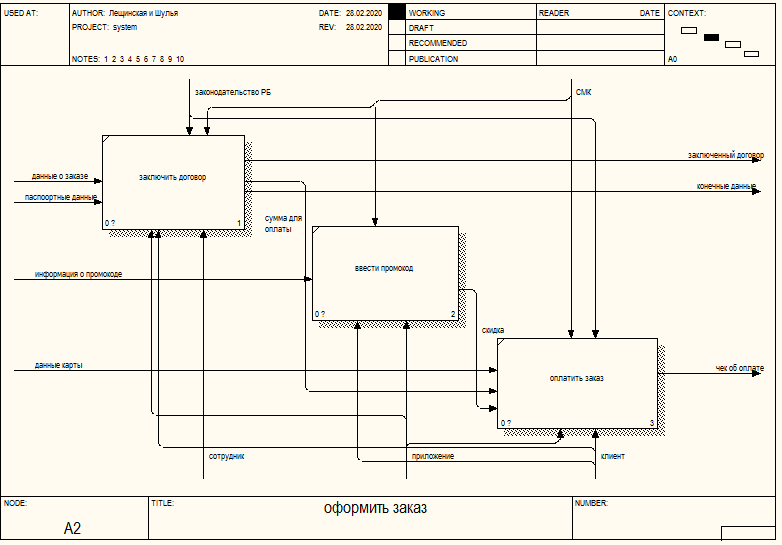


Рисунок 2.13 – Декомпозиция процесса «Оформить заказ»

На рисунке 2.14 показана декомпозиция процесса «Обслужить банкет». Она включает в себя блоки «Перевести гонорар на счёт работников» и «Перевести на счёт деньги за аренду».

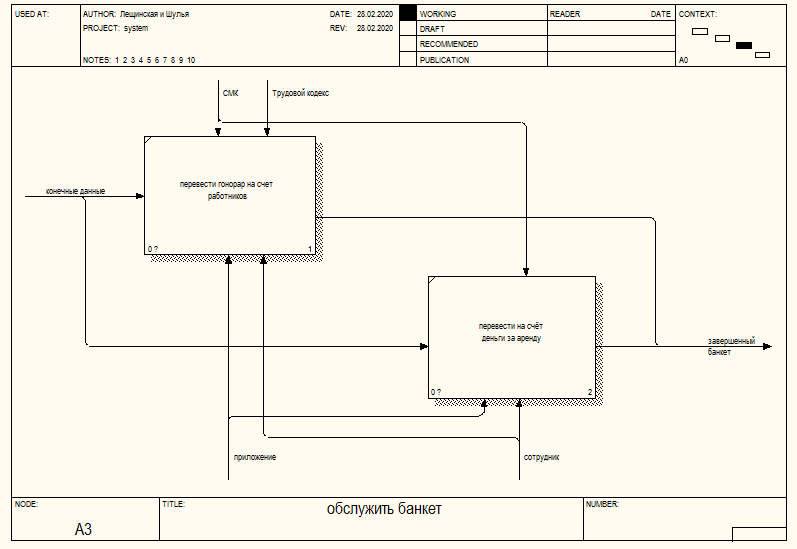


Рисунок 2.14 – Декомпозиция процесса «Обслужить банкет»

Процесс составления отчёта происходит аналогично с этим процессом в системе «Как есть». Ключевым различием между двумя системами является то, что система «Как должно быть» более гибкая, экономит время пользователя, более удобна для работников компании, т.к. не требует длительного оформления документов и ожидания оплаты.

В следующей главе будет рассмотрена спецификация вариантов использования системы.

# сПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Диаграмма вариантов использования является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. В свою очередь вариант использования – это спецификация функций, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с актером. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий [5].

На рисунке 3.1 показана диаграмма вариантов использования для разрабатываемого веб-приложения.

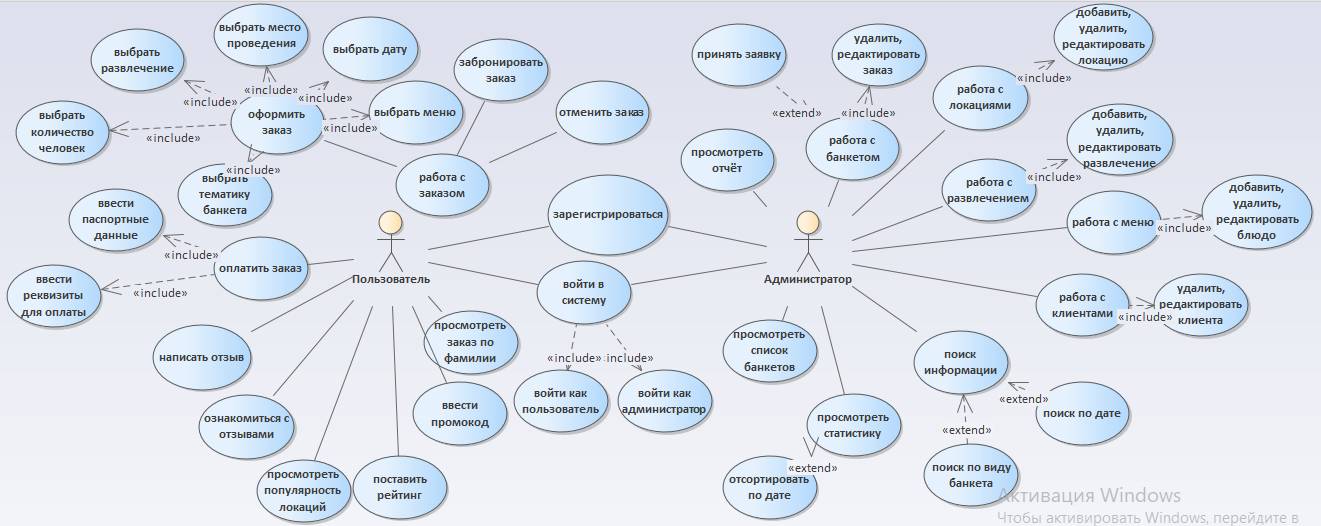


Рисунок 3.1 – Диаграмма вариантов использования

На данной диаграмме отображено какими функциями будут обладать пользователь и администратор. Пользователь сможет выбирать интересующие его пункты, связанные с оформлением банкета, иметь возможность оформить и оплатить договор онлайн, просмотреть популярность локаций, просмотреть свой заказ, предварительно введя свою фамилию, воспользоваться скидкой, введя промокод, написать отзыв, просмотреть список отзывов, оценить данные, относящиеся к банкету, а также забронировать банкет и отменить его.

Администратор в свою очередь может работать с данными пользователя, удалять и редактировать их, может проводить поиск информации по выбранным критериям, просматривать статистику, используя сортировку, работать с данными оформленных заказов, просматривать сформированный за определенный период времени отчёт, принимать заявку на заказ, а также добавлять новые данные связанные с проведением банкета.

# Информационная модель системы

Согласно требованиям, база данных должна быть приведена к третьей нормальной форме.

Первая нормальная форма. Сущность находится в первой нормальной форме тогда и только тогда, когда все атрибуты содержат атомарные значения. Среди атрибутов не должно встречаться повторяющихся групп, т. е. несколько значений для каждого экземпляра.

Вторая нормальная форма. Сущность находится во второй нормальной форме, если она находится в первой нормальной форме и каждый не ключевой атрибут полностью зависит от первичного ключа (не должно быть зависимости от части ключа). Вторая нормальная форма имеет смысл только для сущностей, имеющих сложный первичный ключ.

Третья нормальная форма. Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и никакой не ключевой атрибут не зависит от другого не ключевого атрибута (не должно быть взаимозависимости между не ключевыми атрибутами) [6].

На рисунке 4.1 изображена схема базы данных системы по учету и регистрации заказов клиентов на проведение банкетов.

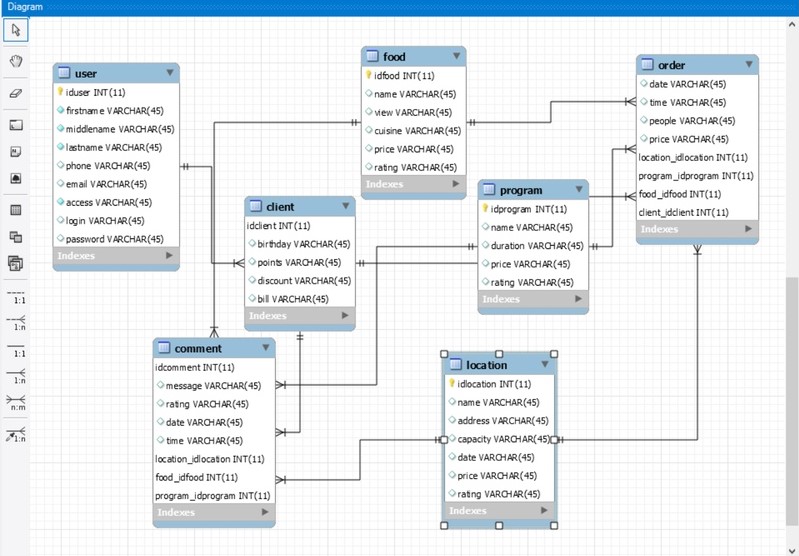


Рисунок 4.1 – Схема базы данных

В процессе разработки были выделены 7 основных сущностей:

* пользователь;
* клиент;
* заказ;
* еда;
* комментарий;
* местоположение;
* программа.

У каждой таблицы имеется свой первичный ключ, а также ключи связей с другими таблицами, благодаря которым имеется возможность получения необходимых данных.

Таблица «order» – это таблица заказа. В неё входят такие поля, как дата «date» типа VARCHAR (45), время «time» типа VARCHAR (45), кличество людей «people» типа VARCHAR (45), сумма «price» типа VARCHAR (45), id местоположения «location\_idlocation» типа INT(11), id программы «program\_idprogram» типа INT(11), id еды «food\_idfood» типа INT(11), id клиента «client\_idclient» типа INT(11).

Таблица «location» – это таблица местоположения банкета. В неё входят поля: ключ местоположения «idlocation» типа INT(11), название «name» типа VARCHAR (45), адрес «address» типа VARCHAR (45), вместимость «capacity» типа VARCHAR (45), дата «date» типа VARCHAR (45), стоимость «price» типа VARCHAR (45), рейтинг «rating» типа VARCHAR (45).

Таблица «program» – это таблица развлечений. Она включает в себя такие поля: ключ программы «idprogram» типа INT(11), название развлечения «name» типа VARCHAR (45), продолжительность «duration» типа VARCHAR (45), стоимость «price» типа VARCHAR (45), рейтинг «rating» типа VARCHAR (45).

Таблица «food» – это таблица с информацией о меню. В неё входят такие поля, ключ еды «idfood» типа INT(11), название блюда «name» типа VARCHAR (45), внешний вид «view» типа VARCHAR (45), кухня «cuisine» типа VARCHAR (45), стоимость «price» типа VARCHAR (45), рейтинг «rating» типа VARCHAR (45).

Таблица «comment» – это таблица отзывов. В неё входят такие поля, как документ «document» типа INT(11), текст «message» типа VARCHAR (45), дата «date» типа VARCHAR (45), рейтинг «rating» типа VARCHAR (45), время «time» типа VARCHAR (45), ключ местоположения «location\_idlocation» типа INT(11), ключ программы «program\_idprogram» типа INT(11) и ключ блюда «food\_idfood» типа INT(11).

Таблица «client» – это таблица личных данных заказчика. В неё входят такие поля, как ключ клиента «idClient» типа INT(11), дата дня рождения «birthday» типа VARCHAR (45), количество бонусных очков «points» типа VARCHAR (45), размер скидки «discount» типа VARCHAR (45), чек «bill» типа VARCHAR (45).

Таблица «user» – это таблица пользователей. В неё входят такие поля, как id пользователя «idUser» типа INT(11), имя «firstname» типа VARCHAR (45), второе имя «middlename» типа VARCHAR (45), фамилия «lastname» типа VARCHAR (45), контактный телефон «phone» типа VARCHAR (45), почта «email» типа VARCHAR (45), адрес «address» типа VARCHAR (45), логин «login» типа VARCHAR (45), пароль «password» типа VARCHAR (45).

Связи сущностей расставлены в таком виде, т.к. для того, чтобы при написании отзыва в системе было понятно о каком месте проведения, меню и программе идет речь, аналогично происходит и при оформлении заказа: должно быть понятие с какими данными происходит работа.

Далее перейдем к более детальному моделированию системы.

# ОБОСНОВАНИЕ ВЫБОРА КОМПОНЕНТОВ И ТЕХНОЛОГИЙ

## Диаграмма последовательности

На рисунке 5.1 представлена диаграмма последовательности для процесса авторизации в системе.

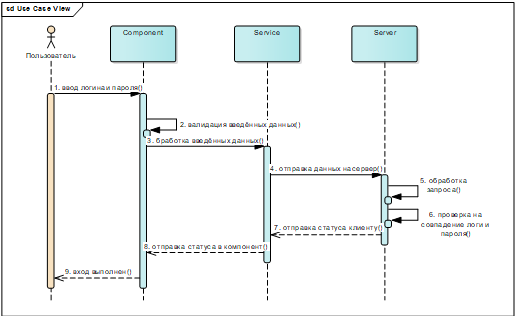
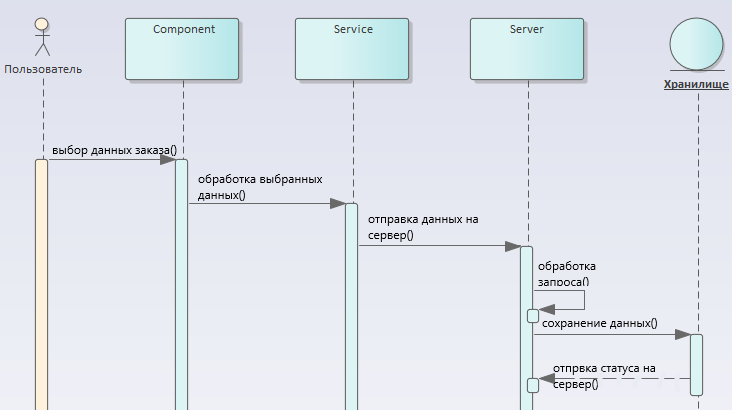
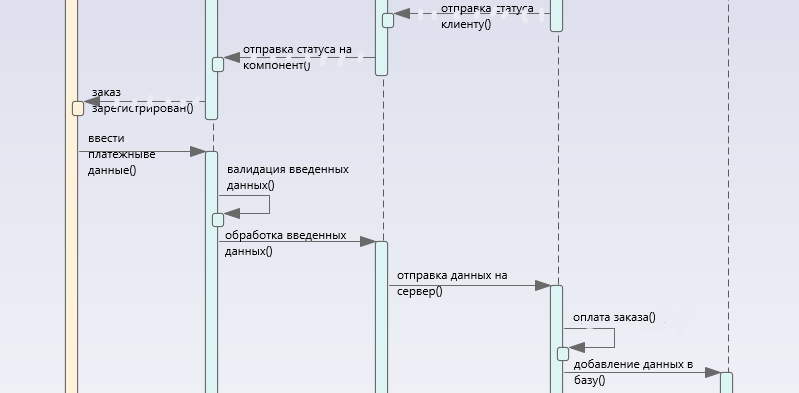


Рисунок 5.1 – Диаграмма последовательности для процесса авторизации

Диаграмма последовательностей отображает временные особенности передачи и приема сообщений между объектами. Время на диаграмме развивается сверху вниз, а жизненный цикл горизонтально. Фокус управления первоначально находится на актере «Пользователь». На странице авторизации он входит в систему, вводя при этом логин и пароль. Введенные данные отправляются в Component. Проверяется корректность данных. В случае успешного ввода, компонент вызывает функцию сервиса, который отправляет данные на сервер. Сервер проверяет логин и пароль на совпадение в базе данных и возвращает статут пользователя обратно сервису. Сервис возвращает компоненту, который выполняет роутинг в случае успешной авторизации.





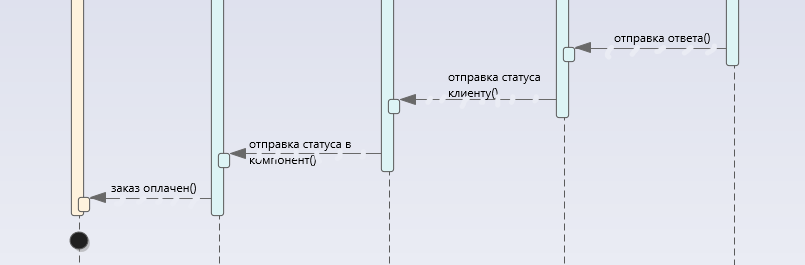


Рисунок 5.2 – Диаграмма последовательности для оформления и оплаты заказа

На рисунке 5.2 показана диаграмма последовательностей для оформления и оплаты заказа. Сначала пользователь выбирает интересующие его данные, затем происходит передача их на компонент и сервер, данные сохраняются в базу. После этого пользователю приходит сообщение об успешной регистрации заказа или об ошибке.

Если оформление прошло успешно, то пользователю предлагается ввести данные для оплаты, они передаются на компонент, затем на сервис и сервер, на нем происходит оплата заказа, после чего данные сохраняются в базу. В конце пользователю приходит сообщение об успешной оплате или ошибке.

## Диаграмма компонентов

Диаграмма компонентов ‒ показывает взаимосвязи между модулями (логическими, физическими), из которых состоит моделируемая система. Основной тип сущностей на диаграмме компонентов – компоненты и интерфейсы, посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

* реализации между компонентами и интерфейсами (компонент реализует интерфейс);
* зависимости между компонентами и интерфейсами (компонент использует интерфейс).

На рисунке 5.3 представлена диаграмма компонентов, которая обуславливает архитектуру разрабатываемой системы.

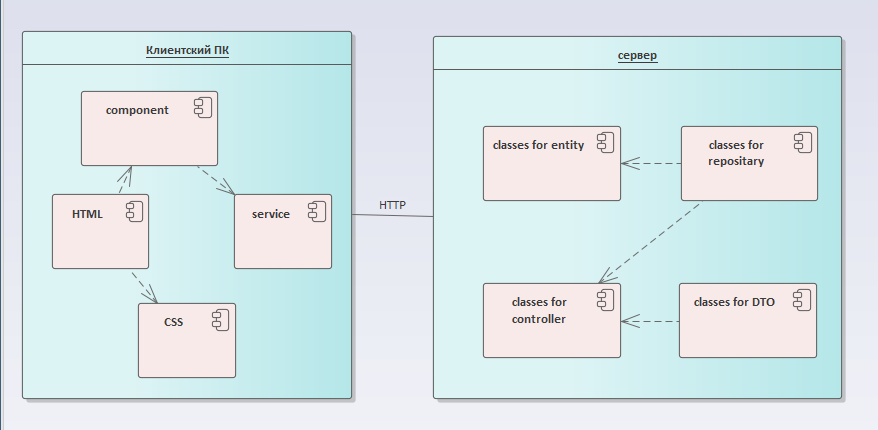


Рисунок 5.3 – Диаграмма компонентов

В следующем подразделе будет описана диаграмма развертывания.

## Диаграмма развертывания

Диаграмма развертывания наряду с отображением состава и связей элементов системы показывает, как они физически размещены на вычислительных ресурсах во время выполнения. Таким образом, на диаграмме размещения, по сравнению с диаграммой компонентов, добавляется два типа сущностей: артефакт, который является реализацией компонента и узел (может быть как классификатор описывающий тип узла, так и конкретный экземпляр), а также отношение ассоциации между узлами, показывающее, что узлы физически связаны во время выполнения.

Процесс создания диаграммы состоит из двух этапов. Первый, основное внимание уделяется на узлы и соединения. Второй, распределение экземпляров артефактов по экземплярам узлов или же артефактов по узлам.

На рисунке 5.4 представлена диаграмма развертывания.

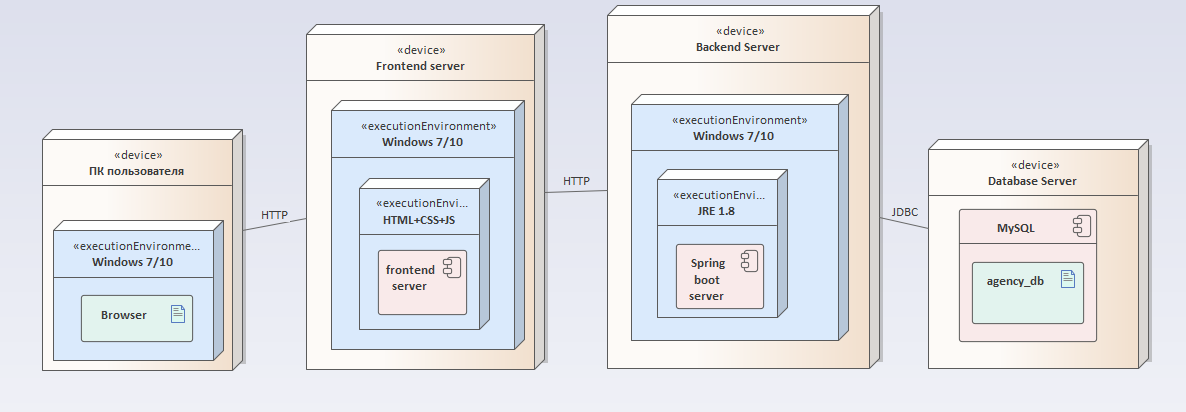


Рисунок 5.4 – Диаграмма развертывания

В следующем подразделе будет подробно описано обоснование выбора технологий.

## Обоснование выбора технологий

Для успешной работы данной системы используются следующие технологии:

* Hibernate/JPA;
* Spring MVC;
* Spring Security;
* Bootstrap;
* jQuery.

Hibernate – это ORM фреймворк для Java с открытым исходным кодом. Эта технология является крайне мощной и имеет высокие показатели производительности.

Hibernate создаёт связь между таблицами в базе данных и Java-классами и наоборот. Это избавляет разработчиков от огромного количества лишней, рутинной работы, в которой крайне легко допустить ошибку и крайне трудно потом её найти [7].

Целью SpringMVC Framework является поддержка в Spring архитектуры модель-представление-контроллер. Spring обеспечивает готовые компоненты, которые могут быть использованы (и используются) для разработки веб-приложений.

Главной целью MVC является разделение объектов, бизнес-логики и внешнего вида приложения. Все эти компоненты слабо связаны между собой и при желании мы можем изменить, например, внешний вид приложения, не внося существенные изменения в остальные два компонента.

Spring Security - это фреймворк, который сфокусирован на обеспечение как аутентификации, так и авторизации в Java-приложениях. Как и все Spring проекты, настоящая сила Spring Security в том, что он может быть легко дополнен нужным функционалом [8].

Для работы с графическим оформлением используются Bootstrap и jQuery.

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

JQuery - быстрая, небольшая и многофункциональная библиотека JavaScript. Она делает такие вещи, как обход и манипулирование документами HTML, обработка событий, анимация и Ajax намного проще с простым в использовании API, который работает во множестве браузеров [9].

В качестве базы данных используется MySQL.

MySQL - это система управления реляционными базами данных. В реляционной базе данных данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных [10].

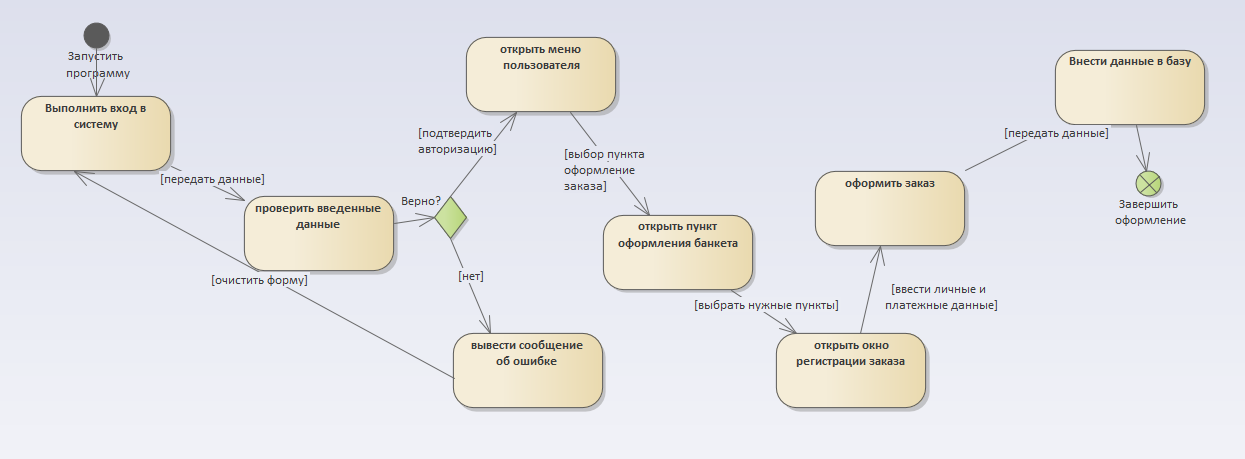
В качестве сервера для запуска приложений используется Tomcat.

Tomcat — контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Реализует спецификацию сервлетов и спецификацию JavaServer Pages (JSP). Написан на языке Java.

Tomcat позволяет запускать веб-приложения, содержит ряд программ для самоконфигурирования [11].

В следующей главе будут описаны модели представления системы.

# МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ



# Руководство пользователя

# Результаты тестирования разработанной системы

# Заключение

# Список используемых источников

1. Оформление банкетов – [Электронный ресурс] – Режим доступа: <https://www.retail-loyalty.org/expert-forum/r-keeper-kak-zarabotat-na-banketakh/;>
2. Учёт заказов – [Электронный ресурс] – Режим доступа: <https://краснаяарка.рф/article/docoformlenie>;
3. Оформление предварительного заказа – [Электронный ресурс] – Режим доступа: <https://www.gb.by/aktual/gosudarstvennoe-regulirovanie/oformlenie-predvaritelnykh-zakazov-v-obs>;
4. Нотация IDEF0 – [Электронный ресурс] – Режим доступа: <https://www.businessstudio.ru/wiki/docs/current/doku.php/ru/csdesign/bpmodeling/idef0>;
5. Диаграмма вариантов использования – [Электронный ресурс] – Режим доступа: [https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12\_2](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2%20) ;
6. База данных – [Электронный ресурс] – Режим доступа: <https://office-menu.ru/uroki-sql/51-normalizatsiya-bazy-dannykh>;
7. Бауэр К., Кинг Г., Грегори Г., Java Persistence API и Hibernate/ пер. С англ. Д.А. Зинкевича; под науч. Ред. А.Н. Киселева – М.: ДМК Пресс. 2017. -632 с.: ил.
8. Лонг Джош, Бастани Кеннет, Java в облаке. Spring Boot, Spring Cloud, Cloud Foundry. – СПб.: Питер, 2019. – 624 с.: ил.
9. JQuery: описание – [Электронный ресурс] – Режим доступа: <https://code.tutsplus.com/ru/tutorials/what-is-jquery--cms-26232>;
10. Описание MySQL– [Электронный ресурс] – Режим доступа: <http://www.webmasterwiki.ru/MySQL>
11. JQuery: описание – [Электронный ресурс] – Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/427774>

# Приложение А Листинг программы