

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 кафедры)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере
(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка элементов информационной системы заказа транспорта
для сотрудников компаний»

Студент	Н.С. Чугунов	
	(И.О. Фамилия)	(личная подпись)
Руководитель	Е.В. Панюкова	
	(И.О. Фамилия)	(личная подпись)
Консультанты	И.Ю. Усатова	
	(И.О. Фамилия)	(личная подпись)

Допустить к защите

Заведующий кафедрой	к.т.н., доцент, А.В. Очеповский	
	(степень, звание, И.О. Фамилия)	(личная подпись)

«_____» _____ 20____ г.

Тольятти 2018

Аннотация

Выпускная квалификационная работа посвящена вопросу автоматизации заказа транспорта для сотрудников компаний. **Актуальность выбранной темы** определяется необходимостью разработки элементов информационной системы, направленной на автоматизацию процессов подачи, обработки, а также согласования заявок сотрудников по мере необходимости.

Объектом исследования выступает деятельность компаний, направленная на заказ автотранспорта для её сотрудников. **Предметом исследования** – автоматизация процессов сбора, хранения, а также обработки данных, используемых в данной предметной области.

Бакалаврская работа состоит из введения, трёх глав основной части, заключения и списка используемой литературы.

Во введении определяется актуальность выбранной темы, обозначаются объект и предмет исследования, а также определяются его цели и задачи.

В первой главе производится анализ бизнес-процессов предприятия, протекающих в рассматриваемой предметной области и требующих автоматизации. Во второй главе, описывается процесс непосредственного проектирования информационной системы, подробно описываются её спецификации. Третья глава посвящена непосредственной технической реализации компонентов системы, рассмотрению их полученного варианта.

В заключении, подводятся итоги, определяются достоинства и недостатки реализованного продукта, обозначаются дальнейшие планы по его развитию.

Результатом ВКР являются разработанные элементы информационной системы, направленной на увеличение оперативности и унификацию процессов подачи, обработки и согласование заявок на заказ транспорта.

В работе представлено 6 таблиц, 38 рисунков, 3 листинга с фрагментами исходного кода, 2 приложения, список использованных источников содержит 28 источников. Общий объем выпускной квалификационной работы составляет 86 страниц.

ABSTRACT

The topic of the given graduation work is “The Employee’s Transportation Ordering Information System Components Development”. This graduation work is devoted to companies business processes automation.

The object of this graduation work is companies’ activities, which aimed to ordering and providing transport for employees. The subject of this graduation work is automation of these processes. The main goal of this project is to design and to develop architecture and functional components for company employee’s transportation ordering information system. The proposed software product aims to increase the efficiency of business processes in this subject area and reduce time costs of employees and company owners.

Explanatory note is divided into three chapters, each of which investigates a specific aspect of development process. In the first chapter, we analyze existing business processes flow and build “AS-IS” conceptual model. After that, we describe and substantiate needs for automation, offer a solution, determine software product requirements and build “TO-BE” conceptual model.

The second chapter deals with functional, logical and domain object models of the system. We give full coverage to its specifications and principles of work.

The third chapter reasonably chooses the stack of technologies for solution implementation, describes direct process of program solution development. After that, at the last paragraphs of explanatory note partially presents implemented functionality.

This graduation work consists of an explanatory note on 86 pages, including 3 chapters, 38 figures, 6 tables, the list of 28 references including 4 foreign sources and 2 appendices with domain model diagram and program code listings.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
Глава 1 Анализ предметной области	10
1.1 Техничко-экономическая характеристика предметной области.....	10
1.1.1 Характеристика предприятий.....	10
1.1.2 Краткая характеристика подразделения и его видов деятельности	12
1.2 Разработка и анализ модели бизнес-процессов «КАК ЕСТЬ»	14
1.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии.....	19
1.4 Анализ существующих разработок на предмет соответствия определенным требованиям	24
1.5 Постановка задачи на разработку АИС.....	26
1.6 Разработка модели бизнес-процессов «КАК ДОЛЖНО БЫТЬ».....	28
Выводы по главе 1	34
Глава 2 Проектирование информационной системы.....	35
2.1 Объектная модель АИС и её описание	35
2.2 Логическая модель АИС и её описание	37
2.3 Функциональная модель АИС и её описание	43
2.4 Информационное обеспечение АИС	44
2.4.1 Описание используемых нормативно-справочных документов	44
2.4.2 Характеристика входной и оперативной информации.....	49
2.4.3 Характеристика выходной информации	51
2.5 Логическая модель данных АИС и её описание	54
Выводы по главе 2.....	55
Глава 3 Техническая реализация компонентов АИС.....	57
3.1 Описание архитектуры АИС.....	57
3.2 Технологическое обеспечение решения	60
3.2.1 Определение набора технологий для реализации серверного программного обеспечения АИС.....	60

3.2.2 Определение набора технологий для реализации клиентского программного обеспечения АИС.....	62
3.2.3 Выбор СУБД и обоснование выбора.....	64
3.3 Техническая реализация компонентов АИС.....	66
3.3.1 Физическая модель БД АИС	66
3.3.2 Описание программных модулей АИС	67
3.3.3 Обобщенное описание процесса сбора, передачи, обработки и выдачи информации.....	69
3.4 Описание реализованной функциональности.....	75
Выводы по главе 3.....	81
ЗАКЛЮЧЕНИЕ	83
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	85
ПРИЛОЖЕНИЕ А Доменная модель предметной области.....	89
ПРИЛОЖЕНИЕ Б Листинг кода сервиса аутентификации	90

ВВЕДЕНИЕ

В современную жизнь общества с каждым днём всё глубже и глубже проникают информационные технологии, становясь для большинства людей чем-то повседневным, доступным, привычным, и, даже, в некоторых случаях, неотъемлемым. Сегодня, сфера информационных технологий развивается сверхбыстрыми темпами: разрабатываются новые операционные системы, корпоративные решения, пакеты прикладных программ, мобильные приложения и утилиты, различные системы искусственного интеллекта и нейросети, приносящие огромную пользу как простым пользователям, так и бизнесу в решении своих повседневных задач различной сложности и масштабов. Можно уверенно сказать, что информационные технологии определяют будущее, задают вектор дальнейшего развития. Именно по этой причине, программная инженерия, на сегодняшний день, является одной из наиболее широкомасштабных сфер человеческой деятельности, а специалисты данной области – особо востребованными.

Рассматриваемый в данной выпускной квалификационной работе процесс разработки направлен на создание элементов информационной системы, автоматизирующей процессы подачи, обработки и акцептации заявок на заказ автотранспорта сотрудниками различных компаний, а также формирования отчётной документации, по мере возникновения такой необходимости.

Объектом исследования выступает деятельность компаний, направленная на организацию процесса предоставления автотранспорта для своих сотрудников.

Предметом исследования является автоматизация процессов сбора, хранения и обработки данных, используемых в рассматриваемой предметной области, а также получения отчётной документации на основании этих данных, при возникновении такой потребности.

Целью выпускной квалификационной работы является разработка элементов автоматизированной информационной системы заказа автотранспорта для сотрудников компаний. Целесообразность разработки,

распространения и внедрения данной информационной системы в ИТ-инфраструктуры компаний обуславливается возможным снижением трудозатрат путем повышения оперативности сбора и обработки данных, уменьшением вероятности допущения ошибок при работе с ними, а также хранения их в структурированном и консистентном виде.

Для достижения цели, поставленной в рамках данной выпускной квалификационной работы, необходимо выполнить ряд **задач**:

- проанализировать учебную, учебно-методическую, а также научно-техническую литературу, посвященную теме разработки автоматизированных информационных систем;
- произвести анализ рабочих процессов деятельности, направленной на заказ транспорта для сотрудников типичной компании, построить их концептуальные модели;
- проанализировать рынок программных продуктов в данной области на предмет наличия аналогов, определить достоинства и недостатки каждого из них;
- на основании результатов первичного анализа, определить и обосновать актуальную необходимость в разработке, распространении и внедрении реализованной автоматизированной системы, а также выработать требования к её проекту;
- обоснованно выбрать архитектурное решение, наиболее подходящее для реализации программного продукта;
- спроектировать хранилище данных информационной системы, его доменную модель;
- произвести обоснованный выбор средств реализации автоматизированной информационной системы;
- реализовать физическое хранилище данных системы посредством выбранных технических средств;
- реализовать серверное приложение системы посредством выбранных технических средств;

- разработать клиентское приложение системы посредством выбранных технических средств;
- протестировать работоспособность и адекватность поведения реализованного программного решения при различных входных данных посредством реализации и выполнения «дымовых», модульных и интеграционных тестов;
- сделать заключение о проделанной работе, описать достоинства и обозначить дальнейшие планы развития программного продукта.

Содержание работы включает в себя введение, три главы, заключение, список использованных источников, а также приложения.

Первая глава работы посвящена анализу типовой структуры предприятий и их бизнес-процессов, направленных на заказ автотранспорта для их сотрудников. В главе представлена концептуальная модель, наглядно отражающая текущее и предлагаемое состояния рассматриваемой предметной области, обосновывается актуальность необходимости разработки автоматизированного решения, приводятся результаты анализа рынка программных продуктов на наличие аналогичных систем. После этого, обозначается цель разработки, а также определяется назначение и вырабатываются требования к новому автоматизированному решению.

Во второй главе приводится поэтапный процесс проектирования компонентов автоматизированной информационной системы: представляются её объектная, функциональная и логическая модели, описывается её информационное обеспечение, а также приводится логическая модель хранилища данных.

Третья глава описывает процесс непосредственной технической реализации решения: архитектурные принципы, положенные в основу проекта, набор технологий для физической реализации его компонентов, а также их отладки. Глава содержит описание реализованных программных модулей с примерами программного кода, механизмов их взаимодействия. Наконец, завершает данную главу параграф, в котором обозначаются основные

функциональные характеристики реализованных компонентов системы, приводятся их частичные визуальные представления.

В заключении подводятся итоги выполненной работы, описываются достоинства и обозначаются планы дальнейшего развития реализованного программного продукта.

Результатом выполнения поставленной цели и определенных задач, должен являться протестированный и задокументированный программный продукт, удовлетворяющий всем предъявленным к нему требованиям.

Глава 1 Анализ предметной области

1.1 Техничко-экономическая характеристика предметной области

1.1.1 Характеристика предприятий

Проектирование информационных систем всегда начинается с анализа предметной области, подлежащей дальнейшей автоматизации, описания ее специфики и «узких моментов». Грамотно и полно описанная предметная область является отправной точкой, главным источником функциональных требований к программному продукту, определяемых в дальнейшем.

Поскольку, предполагается, что разрабатываемая информационная система заказа автотранспорта не будет предназначаться для какой-то одной, определенной компании, будет иметь широкий спектр использования, а организации иерархического вида считаются наиболее распространенными, рассматриваться будет именно данный тип организационной структуры предприятий.

Управленческие структуры на большинстве современных предприятий базируются на принципах, определенных еще в начале XX века. Наиболее полная и точная формулировка данных принципов – концепция рациональной бюрократии, была представлена немецким социологом Максом Вебером. Она подразумевает под собой следующее:

- принцип иерархичности уровней управления, подразумевающий подконтрольность и подчинение нижестоящего уровня вышестоящему;
- соответственно, вытекающий из принципа иерархичности принцип соответствия полномочий и ответственности каждого из сотрудников занимаемому им месту в иерархии;
- принцип функционального разделения труда, основанный на специализации сотрудников;
- принцип формализации, а также стандартизации трудовой деятельности, обеспечивающий однородность выполняемых обязанностей, а также скоординированность выполняемых задач;

– принцип квалификационного отбора, подразумевающий найм и увольнение с работы в строгом соответствии с квалификационными требованиями [19].

Собственно, построенную на данных принципах структуру, принято называть иерархической. Схематичное представление внутреннего устройства типовых иерархически организованных компаний, представлен на рисунке 1.1.



Рисунок 1.1 – Иерархическая организационная структура предприятий

Как можно видеть на представленной схеме, структура компаний, организация которых построена по иерархическим принципам, имеет древовидную форму, размеры которой могут быть абсолютно любыми - как в «ширину», подразумевая подразделения, так и в «глубину», подразумевая уровни управления. Говоря в целом, можно сказать, что иерархическая структура не может считаться хуже или лучше других типов организационных систем. В основе оценки эффективности применения любой организационной структуры лежит ее соответствие целям и задачам, преследуемым компанией, а также возможность эффективной организации рабочих и бизнес-процессов. Построение иерархии управления позволяет обеспечивать параллельно

выполнение независимых друг от друга процессов на одном уровне под общим контролем более высокого руководства.

Иерархическая структура имеет большое количество различных подтипов, однако, суть её применения и использования, в целом, не меняется – имеются конечные исполнители, работники, подчиняющиеся руководству, которые, в свою очередь, имеют свое непосредственное руководство, и так, по возрастающей, вплоть до генерального директора предприятия. Рассматриваемая в данной работе информационная система ориентирована именно на предприятия подобного типа.

Определив основные понятия иерархической структуры предприятий, перейдем к описанию предметной области – типовых процессов, направленных на предоставление сотрудникам компаний автотранспорта, а также заинтересованных и задействованных в этих процессах лиц.

1.1.2 Краткая характеристика подразделения и его видов деятельности

В настоящее время, среди предприятий различных сфер и масштабов деятельности, достаточно широко распространенной практикой является предоставление корпоративного транспорта для своих сотрудников. Как правило, транспорт бывает необходим для посещения точек интереса – мест, непосредственно связанных с деятельностью компании и предоставляющих для нее определенную значимость. Примерами точек интереса могут служить офисы компаньонов, банки, различные государственные организации, а также, если речь ведется о командированных сотрудниках, точкой интереса, например, может быть офис компании в городе – пункте назначения командированного и т.д. Кроме того, многие организации зачастую организуют доставку своих сотрудников до места жительства в вечернее время, и от места жительства до места работы в утреннее. Они могут делать это как собственным транспортом, так и привлекая сторонние компании, оказывающие услуги пассажирских перевозок, если собственного транспорта не имеется.

Как правило, каждая компания имеет в своем штате офис-менеджера или, иными словами, делопроизводителя – человека, ответственного за встречу

посетителей офиса, организацию совещаний, митингов и брифингов, снабжение сотрудников всем необходимым для их полноценной работы, контролирующего расходы на офисные нужды, принимающего звонки, обрабатывающего почту, а также распределяющего документацию по подразделениям. Исходя из этой позиции, далее по тексту данной работы, лицом, ответственным за обработку заявок будет считаться именно этот сотрудник.

При возникновении потребности в автотранспорте, сотрудники могут осуществлять заказ транспорта самостоятельно, или оставлять заявки офис-менеджеру. Они имеют возможность сделать это как в устной форме, так и путем отправки e-mail сообщения или заполнения специального бланка, если речь идет о групповой доставке сотрудников. После оформления заявки, она регистрируется в журнале, форма которого, в целом, имеет типовую структуру документов данного вида, с определяемыми непосредственно организациями деталями. После регистрации, для исполнения заявки подбирается компания, оказывающая транспортные услуги, с ней осуществляется связь, а затем заказывается транспорт в соответствии с предъявленными к поездке требованиями, о чем сотрудники персонально уведомляются, устно, в форме звонка или, например, e-mail сообщения.

Существуют три способа организации оплаты поездок для сотрудников:

- 1) по корпоративной банковской карте, на которую переводятся средства, выделяемые на поездки на определенный период;
- 2) из собственных средств сотрудника, которые позже компенсируются компанией при предоставлении соответствующих документов;
- 3) по корпоративному договору, в рамках которого компания получает единый счёт к оплате за все совершенные поездки в конце каждого месяца [21].

Иногда, по тем или иным причинам, могут возникать такие ситуации, когда заказ оформляется только с согласования вышестоящего руководства. Причиной возникновения такой ситуации может служить, например,

потенциально внушительная стоимость поездки, высокая частота поездок одного сотрудника, и т.д, а результатом согласования – как акцептация и, как следствие, оформление заказа транспорта, так и ее отклонение, а, следовательно прекращение жизненного цикла её обработки.

Вся информация, фиксируемая в результате вышеперечисленных действий и непосредственно касающаяся финансовой стороны, может предоставляться в бухгалтерию для составления отчетной документации, а также компенсаций затрат сотрудников, в случае оплаты услуг из собственных средств и предоставления ими соответствующих платежных документов, подтверждающих данный факт.

Таким образом, можно заключить, что рабочий поток данных бизнес-процессов подразумевают выполнения большого спектра задач, многие из которых, очевидно, требуют однообразной рутинной работы, отнимающей большое количество времени. Очевидно, что в целях уменьшения трудозатрат на выполнение этих задач, необходимо разработать информационную систему, позволяющую обеспечить автоматизацию бизнес-процессов, направленных на заказ и предоставление корпоративного транспорта для сотрудников компаний.

Выводы по параграфу 1.1

В параграфе 1.1 была рассмотрена традиционная структура организаций и принципы, на которых она базируется. Помимо этого, была дана краткая характеристика основных лиц, участвующих в процессах, направленных на заказ и предоставление корпоративного автотранспорта. Данная информация является исходным материалом для дальнейшего построения концептуальных моделей, определения и обоснования необходимости разработки автоматизированного решения, а также выработки требований к конечному программному продукту.

1.2 Разработка и анализ модели бизнес-процессов «КАК ЕСТЬ»

Построение модели «КАК ЕСТЬ» («AS-IS») подразумевает создание диаграммы, функционально описывающей рабочий поток бизнес-процессов

рассматриваемой предметной области на текущий момент. Построение данной модели позволяет систематизировать бизнес-процессы, протекающие в рассматриваемой предметной области, получить их формализованное и наглядное представление. Проведение анализа бизнес-процессов на основании построенной модели «КАК ЕСТЬ» позволяет определить узкие места в ИТ-инфраструктуре компании, а также бизнес-процессы из общего рабочего потока, организованные не оптимально.

Концептуальные модели зачастую строят при помощи различных графических и CASE-средств, значительно упрощающих данный процесс. Функциональная модель «КАК ЕСТЬ», построенная при помощи условно-бесплатного CASE-средства Ramus представлена на рисунке 1.2.

Цель моделирования: анализ протекания бизнес-процессов, направленных на заказ транспорта для сотрудников компаний.

Точка зрения: офис-менеджер.

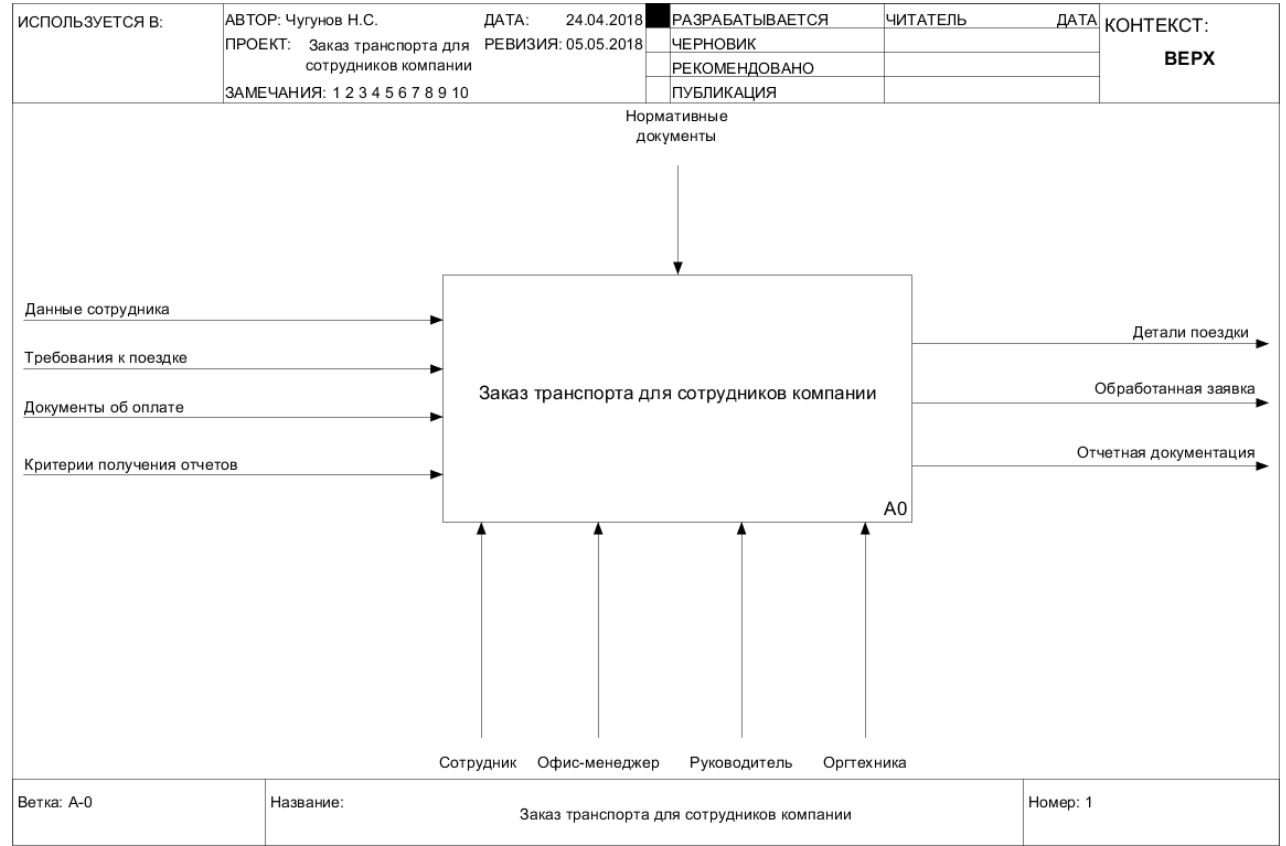


Рисунок 1.2 – Контекстная диаграмма модели «КАК ЕСТЬ»

Как можно видеть на рисунке, входными являются данные сотрудника, требования к поездке, фиксируемые в журнале, документы, подтверждающие оплату, а также критерии получения отчётной документации, определяющие её необходимость и тип. В деятельности, направленной на заказ автотранспорта принимает участие сам сотрудник, имеющий потребность в заказе транспорта, офис-менеджер, принимающий и обрабатывающий заявки, руководитель, с которым, по мере необходимости, заявки согласовываются, а также используется оргтехника. В случае положительного результата согласования (если заявка проходила данный процесс), а также заказа транспорта, на выходе сотрудник получает детали предстоящей поездки – цвет, номер автомобиля и т.д, а сама заявка, как таковая, становится обработанной. Кроме того, на выходе может получаться отчётная документация, формируемая по мере необходимости и соответствующая обозначенным критериям. Деятельность каждого из лиц, участвующих в процессе, регулируется нормативными документами.

После построения контекстной диаграммы, представляющей рассматриваемую предметную область в виде «черного ящика», для более полного описания состава бизнес-процессов, протекающих внутри, на рисунке 1.3 приводится её декомпозиция – разделение на процессы верхнего уровня.

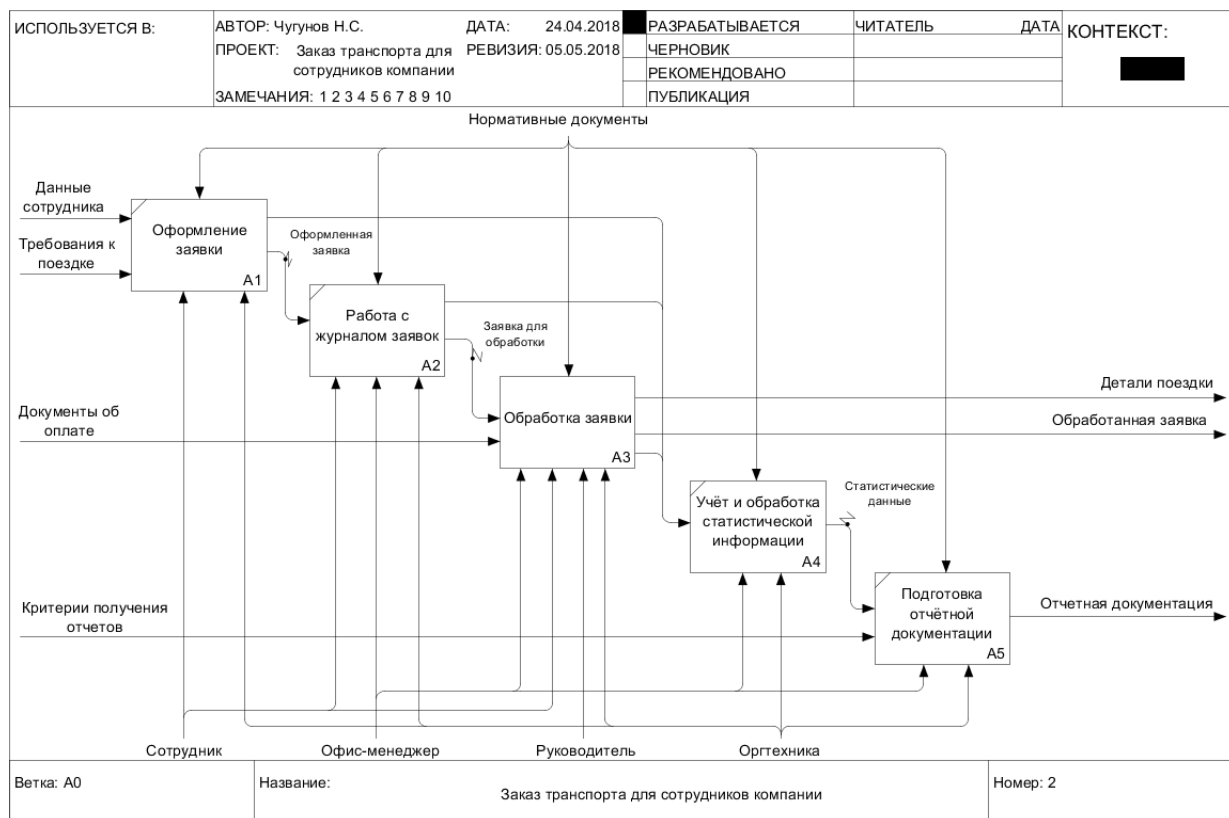


Рисунок 1.3 – Декомпозиция контекстной диаграммы модели «КАК ЕСТЬ»

Основные функциональные блоки, представленные на рисунке 1.3, определяют следующие процессы:

1. Процесс A1 - «Оформление заявки на сотрудника» подразумевает, что какой-либо из сотрудников имеет потребность в автотранспорте. Для этого, он обращается к офис-менеджеру, оставляет запись в журнале заявок, указывая свои данные, а также непосредственные требования к поездке: дата и время, пункт прибытия, необходимый класс автотранспорта и т.д. В том случае, если транспорт необходим срочно, так же делается соответствующая пометка. На выходе из данного процесса, получается оформленная заявка, поступающая в журнал.

2. Процесс A2 - «Работа с журналом заявок» подразумевает ведение журнала заявок, а также непосредственный выбор заявок для обработки. На вход, данный процесс всегда получает оформленные сотрудниками заявки. На выходе из данного процесса, получается заявка, выбранная для обработки, а также часть данных, необходимых для статистического учёта.

3. Процесс АЗ - «Обработка заявки» подразумевает проведение ряда действий, направленных на удовлетворение потребностей сотрудника в определенном виде транспорта. На выходе, данный процесс имеет закрытую заявку, часть данных, необходимых для статистического учёта, обработанную заявку, а также, детали поездки, сообщаемые сотруднику.

4. Процесс А4 - «Учёт и обработка статистической информации» - предполагает агрегацию информации и документов, обрабатываемых на предыдущих уровнях, хранение их для использования в качестве материала для подготовки отчётной документации.

5. Процесс А5 - «Подготовка отчётной документации» - предполагает подготовку отчётной документации на основе имеющейся статистической, агрегированной информации о заявках, заказах и т.д. На вход, данный процесс получает статистическую информацию и критерии формирования отчётов, а на выходе имеет непосредственно оформленную отчётную документацию.

На основании представленных выше схем и их описаний, можно прийти к выводу, что организация рабочего потока бизнес-процессов является неоптимальной и имеет ряд недостатков:

- низкая производительность труда;
- высокая трудоемкость обработки информации;
- несовершенство организации сбора и регистрации исходной информации;
- несовершенство процессов сбора, передачи, обработки, хранения, защиты целостности информации, а также выдачи результатов конечному пользователю.

Учитывая все вышесказанное, можно сделать вывод, что имеется актуальная необходимость в разработке и внедрении автоматизированной информационной системы, что позволит значительно сократить время обработки и получения оперативных данных, повысить степень достоверности

информации, обеспечить более высокую степень ее защищенности и целостности.

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

Следует заметить, что бизнес-процессы подразумевают работу с большим количеством бумажной документации - сбор, хранение и обработка информации о заказах, при отсутствии заключенных договоров с крупными компаниями, предоставляющими своим клиентам технические средства для частичной автоматизации вышеприведенных задач остается ручным - вся работа ведется, как правило, в прикладных приложениях из стандартного набора пакета Microsoft Office, не обеспечивающих должного уровня защищенности, целостности и консистентности данных, оставляя вероятность допущения ошибок или утери данных достаточно высокой. Кроме того, технические средства, предоставляемые крупными таксомоторными компаниями, не решают проблему полностью, поскольку направлены на прямой заказ предоставляемого ими и только ими транспорта, хранение и обработка такими системами данных происходит извне, а расширение функциональности при возникновении потребности является невозможным ввиду отсутствия, открытого и задокументированного прикладного программного интерфейса. Наконец, процессы, направленные на предоставление корпоративного транспорта внутри компании, не являются унифицированными и систематизированными, не имеют единого подхода к их организации. В дополнение к вышесказанному, можно сказать, что, сбор и обработка информации, а также составление отчетов требуют достаточно больших трудозатрат, что также отрицательно сказывается на общей производительности уполномоченных сотрудников.

На основании вышеизложенного, можно прийти к выводу, что в настоящий момент, имеет место актуальная необходимость в разработке системы, подходящей для компаний с иерархической структурой и различными масштабами деятельности, направленной на автоматизацию процессов

отправки сотрудниками заявок на заказ автотранспорта, обработки и при необходимости согласования данных заявок руководством, а также поиска информации и формирования отчетной документации на основании этих данных в случае возникновения такой необходимости.

Формирование требований – термин, используемый для описания процесса определения и документирования ряда требований, предъявляемых к проектируемой информационной системе.

Вполне рациональным классификатором требований могут служить показатели модели FURPS. Аббревиатура FURPS означает следующие категории требований:

1) F - functionality – функциональные требования – являются основным источником для моделирования прецедентов, определяют требуемую функциональность, подлежащую реализации;

2) U - usability – требования к применимости – определяют удобство использования системы её пользователями;

3) R - reliability – требования к отказоустойчивости, ее предсказуемости, восстанавливаемости, а так же надежности системы;

4) P - performance – требования к производительности, использованию ресурсов, масштабируемости и эффективности системы;

5) S - supportability – требования к поддержке, гибкости и расширяемости системы.

Стоит добавить, что требования могут иметь атрибуты, используемые для представления дополнительной описательной информации. Наиболее значимым таким атрибутом является «Приоритет», значение которого определяет важность атрибутированного требования среди их общего числа. Для назначения требованиям приоритетов зачастую используется набор критериев MoSCoW, семантика которого представлена в таблице 1.1.

Таблица 1.1 – семантика набора критериев MoSCoW

Значение атрибута Приоритет	Семантика
1	2
Must have (обязан иметь)	Фундаментальные по отношению к системе

	требования, подлежащие обязательному выполнению
Should have (должен иметь)	Важные требования, которые, однако, могут быть опущены по тем или иным причинам
Could have (мог бы иметь)	Требования, выполняемые при наличии доступных временных ресурсов
Would to have (хотел бы иметь)	Не критичные для выполнения требования, которые могут быть реализованы в последующих версиях системы

При использовании схемы MoSCoW, у каждого из требований есть атрибут «Приоритет», который может принимать одно из значений: М, S, С или W, соответственно.

На основании результатов идентификации предметной области, был выработан перечень наиболее существенных требований к проектируемой системе. Следует отметить, что разработка любой информационной системы представляет собой трудоемкий, длительный процесс, с постепенным наращиванием функциональности, переходом от простого к сложному. Исходя из этого, было принято решение о распределении общей функциональности конечного продукта, а соответственно, и требований к нему по нескольким его версиям. Версия программного продукта, в которой предполагается поставить ту или иную функциональность, описывается одноименным атрибутом «Целевая версия». Перечень сформированных требований представлен в таблице 1.2.

Таблица 1.2 – Перечень требований к информационной системе

№ П/П	Формулировка требования	Приоритет	Целевая версия
1	2	3	4
1	Предоставлять сотруднику возможность создания заявки на заказ транспорта	М – обязан иметь	1.0
2	Предоставлять сотруднику возможность просмотра списка своих заявок	М – обязан иметь	1.0
3	Предоставлять сотрудникам возможность детального просмотра каждой из своих заявок	М – обязан иметь	1.0
4	Предоставлять офис-менеджеру возможность отслеживать и обрабатывать список заявок, а также управлять их жизненным циклом	М – обязан иметь	1.0
5	Предоставлять возможность офис-менеджеру отправлять заявки на согласование с руководством	М – обязан иметь	1.0

6	Предоставлять руководителю возможность отслеживать список заявок на согласование	М – обязан иметь	1.0
7	Предоставлять руководителю возможность акцептовывать или отклонять заявки	М – обязан иметь	1.0
8	Предоставлять сотруднику возможность отслеживать текущий список заявок и их жизненный цикл	М – обязан иметь	1.0
9	Предоставлять сотрудникам и офис-менеджеру возможность совершать отмену заявок	М – обязан иметь	1.0
10	Осуществлять рассылку уведомлений о предстоящих поездках на электронную почту.	М – обязан иметь	1.0
11	Предоставлять руководителю и офис-менеджеру возможность формирования отчетной документации и выгрузки в формате .xlsx	М – обязан иметь	1.0
11	Предоставлять офис-менеджеру оформлять заявки на других сотрудников	М – обязан иметь	1.0
12	Предоставлять механизмы администрирования и конфигурирования системы	М – обязан иметь	1.1
13	Предоставлять офис-менеджеру возможность объединения нескольких заявок в одну	С – мог бы иметь	1.1
14	Предоставлять механизмы регистрации этапов жизненного цикла заявки	W – хотел бы иметь	1.1

Продолжение таблицы 1.2

1	2	3	4
15	Предоставлять механизмы формирования маршрутов поездок с промежуточными точками	W – хотел бы иметь	1.2
16	Осуществлять рассылку уведомлений о поездках на мобильный телефон	W – хотел бы иметь	1.2
Требования к удобству использования			
1	Система должна быть реализована в виде веб-приложения	M – обязан иметь	1.0
2	Пользовательский интерфейс системы должен быть удобен и интуитивно понятен	M – обязан иметь	1.0
3	Система должна иметь удобную навигацию, предоставлять справочную информацию и подсказки	M – обязан иметь	1.0
4	Размер основного шрифта должен быть 14 px	M – обязан иметь	1.0
5	Интерфейс пользователя системы должен иметь приятную, не раздражающую цветовую гамму	M – обязан иметь	1.0
6	Система должна предоставлять возможность переключения интерфейса на вариант для слабовидящих	W – хотел бы иметь	1.2
Требования к надежности			
1	Система должна иметь возможность функционирования 24 часа в сутки, 7 дней в неделю	M – должен иметь	1.0
2	Система должна предоставлять механизмы авторизации и аутентификации пользователей, а также обеспечивать разграничение доступа к функционалу системы путём выдачи определенных прав	M – обязан иметь	1.0
3	Система должна предоставлять механизмы валидации вводимых данных и защиту от атак, взломов и SQL-инъекций	M – обязан иметь	1.0
4	Пароли учётных записей пользователей при записи в базу данных должны хэшироваться	M – обязан иметь	1.0
5	Информация системы должна быть расположена в закрытой БД, доступ к которой должен осуществляться только из программных модулей системы и с использованием учётной записи пользователя	M – обязан иметь	1.0
6	Сервер, на котором расположено веб-приложение, должен иметь резервный канал доступа в сеть Интернет и ИБП, обеспечивающие функционирование системы в течение 1 часа с момента наступления аварийной ситуации	W – хотел бы иметь	1.2

Окончание таблицы 1.2

1	2	3	4
Требования к производительности			
1	Система должна иметь возможность обеспечить бесперебойную одновременную работу как минимум 30 пользователям	М – обязан иметь	1.0
2	Время отклика информационной системы не должно превышать 5 секунд	М – обязан иметь	1.0
Требования к сопровождаемости			
1	Устранение обнаруженных в системе проблем должно быть не дольше 12 часов	М – обязан иметь	1.0
2	Система должна реализовывать механизмы конфигурирования и администрирования	М – обязан иметь	1.1
3	Система должна реализовывать механизмы логгирования для более быстрого определения возможных причин возникновения ошибок, а также отслеживания действий пользователя	М – обязан иметь	1.1

Не исключая возможности существования информационных систем, соответствующих требованиям, определенным выше, следует проанализировать рынок программных продуктов на предмет наличия таковых, а затем, произвести их сравнительную характеристику, с целью обоснования актуальности разработки автоматизированной информационной системы.

1.4 Анализ существующих разработок на предмет соответствия определенным требованиям

Проведение анализа существующих разработок, направленных на автоматизацию рассматриваемой предметной области является одним из обязательных и важных шагов, выполняемых на ранних этапах, поскольку, позволяет определить присутствие на рынке программных продуктов-аналогов проектируемой системы, оценить их достоинства и недостатки, процент соответствия реализуемой ими функциональности предъявленным к проектируемой системе требованиям, определить возможность интеграции нового программного продукта с одним из уже имеющихся аналогов. В конечном итоге, результаты проведения такого анализа однозначно определяют целесообразность проектирования и разработки новой системы.

Сравнительная характеристика функциональных показателей рассматриваемой системы и её аналогов представлена в таблице 1.3.

Таблица 1.3 – Сравнительная характеристика АИС и её аналогов

Название продукта Название показателя	Uber	Gett	Яндекс. Такси	АИС
1	2	3	4	5
Предоставление личного кабинета	+	+	+	+
Контроль расходов	+	+	+	-
Автоматизация заказа	+	+	+	±
Возможность заказа транспорта в различных компаниях, даже не предоставляющих средств автоматизации процесса заказа транспорта	-	-	-	+
Возможность организации корпоративной политики поездок	+	+	+	±
Организация унифицированного процесса заказа транспорта внутри компании	±	±	±	+
Возможность формирования отчётов	+	+	+	+
Возможность внедрения в ИТ-инфраструктуру предприятия	-	-	-	+
Открытый, API (прикладной интерфейс)	-	-	+	+
ИТОГО	5,5	5,5	6,5	7

Исходя из представленной в таблице 1.3 сравнительной характеристики, можно сделать вывод, что аналоги рассматриваемой системы, несмотря на их популярность, удобство использования и ряд преимуществ, имеют также и ряд серьезных недостатков:

- 1) не предназначены для организации унифицированного процесса заказа транспорта внутри компаний, имеют несколько другую направленность;
- 2) не предоставляют возможности интеграции в ИТ-инфраструктуру предприятий;
- 3) не имеют открытого и задокументированного API, позволяющего расширять систему при необходимости – функциональная база программных

продуктов дорабатывается только их непосредственными разработчиками и определяется для всех конечных пользователей, что означает невозможность доработки программного продукта по своему усмотрению и нуждам.

На основании этого, можно сделать окончательный вывод о целесообразности разработки системы, поскольку её использование предполагает организацию единого подхода к описываемым бизнес-процессам, призвана быть абсолютно бесплатной, а также иметь открытый и задокументированный прикладной интерфейс, который позволил бы расширять уже имеющуюся функциональность под специфичные нужды предприятия, его использующего.

1.5 Постановка задачи на разработку АИС

Как уже было отмечено ранее, в настоящее время, процессы оформления и согласования заявок на заказ транспорта, их учёта, а также работы с отчётностью полностью основываются на работе с бумажной документацией. Использование такого подхода является нерациональным и трудозатратным, о чём свидетельствуют следующие факты:

- существующий на данный момент процесс оформления, согласования и обработки заявок на заказ транспорта отнимает у вовлеченных в него сотрудников значительное количество времени;
- вышеописанные процессы не унифицированы, т.е. нет полноценного, единообразного подхода к организации жизненного цикла заявок, а также рабочего потока процессов, направленных на работу с ними;
- с течением времени количество документации растёт, а скорость её обработки – только усложняется, также отрицательно воздействуя на производительность труда;
- данные, хранящиеся и распространяемые в электронном виде (файлы Microsoft Word, Excel и т.д.), зачастую избыточны и неконсистентны, содержат большое количество ошибок, могут быть легко утеряны.

Данные аргументы позволяют прийти к выводу, что в целях достижения более высокого уровня оперативности и производительности труда, показателей качества обработки информации, уровня её защищённости, а также сокращения количества бумажной документации, необходимо разработать информационную систему, направленную на автоматизацию процесса заказа транспорта для сотрудников компаний. Следует принимать во внимание тот факт, что разработка информационной системы представляет собой длительный, итеративный процесс с постепенным наращиванием функциональности, его тестированием, а также исправлением дефектов, выявленных в ходе непосредственного функционирования программного продукта. Поэтому, в настоящий момент необходимым и достаточным является разработка элементов системы, удовлетворяющих определенным требованиям, но, с «заказом на будущее», подразумевающим под собой использование наиболее гибкого архитектурного решения, позволившего бы минимизировать риски при дальнейшем расширении функциональности.

Основным входным и оперативным документом, необходимым для работы системы, является заявка на заказ транспорта, содержащая в себе информацию о сотруднике, испытывающем необходимость в заказе транспорта, а также его требования к поездке – предполагаемую дату и время, а также пункты отправки и назначения. Не исключается возможность того, что в компании, которая будет применять данный программный продукт, разным сотрудникам может быть необходим транспорт различных классов. После заполнения определенной экранной формы, документ должен фиксироваться в системе и обрабатываться лицом, ответственным за данную сферу внутренней деятельности компании. Предполагается, что это будет офис-менеджер, однако, система должна предоставлять возможность организации доступа к этому функционалу любому необходимому сотруднику, путём выдачи ему соответствующих прав. При возникновении необходимости, заявка может отправляться на согласование с вышестоящим руководством, результатом согласования, соответственно, может являться ее акцептация, после которой

заявка продолжает свою обработку, либо же, ее отклонение, после чего заявка должна считаться закрытой. Система должна иметь в своём составе модуль, реализуемый механизмы оповещения всех лиц, участвующих в процессе, при наличии какой-либо важной для них информации.

Помимо основной функциональности, проектируемая автоматизированная система должна обеспечивать поддержку используемой информационной базы, целостности, и защищённости её данных. Соответственно, обязательным требованием к системе является наличие механизмов авторизации и разграничения доступа путём выдачи пользователям учетных записей с соответствующими правами. Также, в целях обеспечения корректности передаваемых вводимых данных, система обязана предоставлять механизмы их валидации.

Архитектурно, система должна быть реализована так, чтобы обеспечить минимизацию рисков, имеющих место быть при её расширении и сопровождении. Система должна иметь механизмы конфигурирования и администрирования. Наконец, поскольку систему предполагается использовать как общедоступное решение, необходимо предоставить удобную систему её развёртывания на машинах конечных пользователей, а также организовать справочную документацию, описывающую основные аспекты и принципы работы с ней.

1.6 Разработка модели бизнес-процессов «КАК ДОЛЖНО БЫТЬ»

Как уже было определено ранее, далеко не во всех компаниях, процесс заказа транспорта для сотрудников является автоматизированным - учёт и отчетная документация ведется посредством стандартного набора офисных приложений. Следует отметить, что данный процесс является достаточно трудозатратным, а также предполагает наличие большого количества избыточных и не всегда консистентных данных, вероятность допущения ошибок так же, присутствует. Проектирование, разработка, распространение и внедрение в эксплуатацию рассматриваемой системы призвано внести изменения в модель бизнес-процессов, существующую в таких компаниях на

данный момент посредством предоставления её конечным пользователям механизмов, позволяющих делегировать часть своих рутинных задач, а также действий данному программному продукту, а также организовать единый подход к организации данных бизнес-процессов.

Для отражения предполагаемого протекания бизнес-процессов после внедрения нового программного решения, была построена концептуальная модель «КАК ДОЛЖНО БЫТЬ» («ТО-BE»), контекстная диаграмма которой представлена на рисунке 1.4.

Цель моделирования: отражение протекания бизнес-процессов, направленных на заказ транспорта для сотрудников компаний после внедрения информационной системы.

Точка зрения: офис-менеджер или иное ответственное лицо.

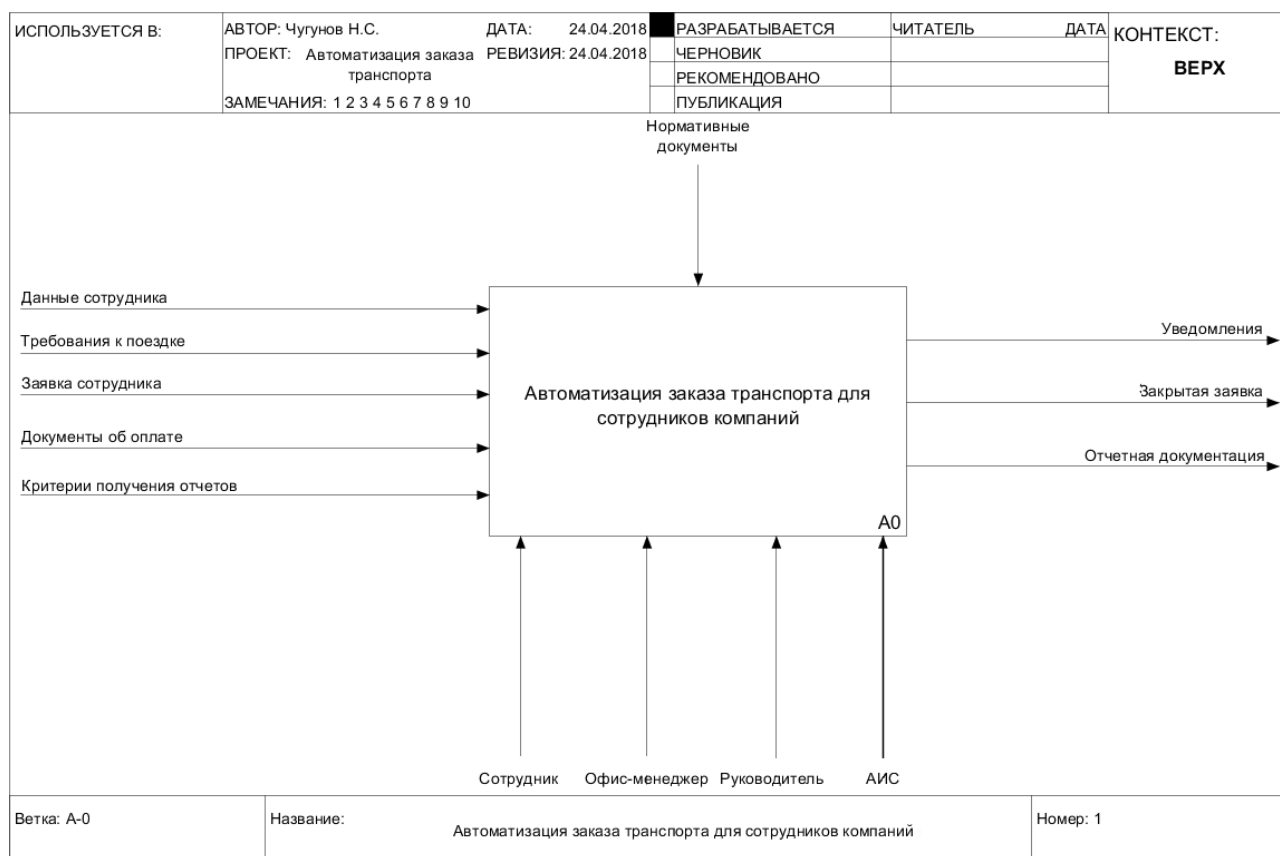


Рисунок 1.4 – Контекстная диаграмма модели «КАК ДОЛЖНО БЫТЬ»

Как можно видеть, на вход процесса «Автоматизация заказа транспорта для сотрудников компаний», поступают данные сотрудника, совершающего

заказ, а также, собственно, требования к поездке, подразумевающие необходимые дату, время, место отправки и прибытия, необходимый тип транспорта и прочие мелкие детали. Это те данные, которые использует офис-менеджер для оформления заявки на какого-либо сотрудника, при возникновении такой необходимости.

Также, на вход непосредственно поступают заявки, отправляемые самими сотрудниками, а также, документы, подтверждающие факт оплаты в тех случаях, когда поездка осуществлялась не за счёт компании, оплачивалась из личных средств сотрудника, и по ним потребуется компенсация затраченных средств. Наконец, система принимает на вход критерии, определяющие формируемую отчётную документацию, получаемую на выходе. Также, на выходе всегда получается закрытая заявка, а любые переходы между этапами жизненного цикла заявки сопровождается отправкой соответствующих уведомлений. Протекание данного процесса регулируется нормативными документами, а к числу исполнителей добавилась автоматизированная информационная система.

Декомпозиция контекстной диаграммы модели «КАК ДОЛЖНО БЫТЬ» представлена на рисунке 1.5. Как можно видеть, рабочий поток на данном уровне представлен следующими четырьмя процессами:

1. Процесс A1 - «Оформление заявки на сотрудника» подразумевает, что какой-либо из сотрудников имеет потребность в автотранспорте, но по тем или иным причинам не может создать заявку самостоятельно, и, соответственно, офис-менеджеру необходимо это сделать. Для этого, он отправляет персональные данные сотрудника, а также обозначает требования к поездке. На выходе из данного процесса всегда получается созданная заявка, часть данных, сохраняемых в качестве статистики и соответствующее уведомление для сотрудника.

2. Процесс A2 - «Работа со списком заявок» подразумевает мониторинг текущего списка созданных, находящихся в работе или обработанных заявок, их статусов, а также непосредственный выбор необходимой заявки для

обработки. На вход, данный процесс всегда получает созданные заявки, как сотрудниками, так и непосредственно офис-менеджером. На выходе, данный процесс имеет заявку, выбранную для обработки, часть данных, сохраняемых в качестве статистики, а также соответствующее уведомление для сотрудника и офис-менеджера.

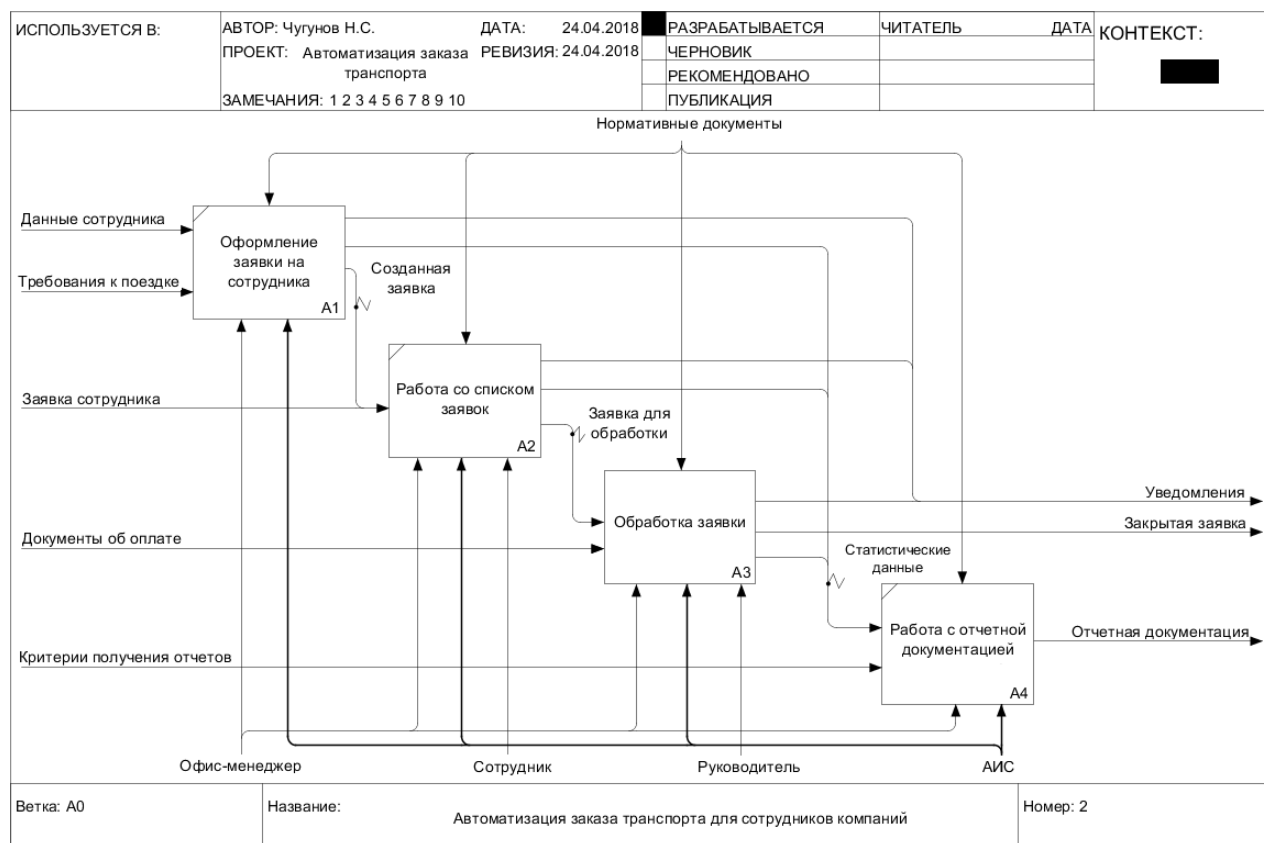


Рисунок 1.5 – Декомпозиция контекстной диаграммы модели
«КАК ДОЛЖНО БЫТЬ»

3. Процесс A3 - «Обработка заявки» подразумевает проведение ряда действий, направленных на удовлетворение потребностей сотрудника в определенном виде транспорта. На выходе, данный процесс имеет закрытую заявку, часть данных, сохраняемых в качестве статистики, а также соответствующие уведомления для заинтересованных лиц;

4. Процесс A4 - «Работа с отчетной документацией» - предполагает генерацию отчётной документации на основе имеющейся информации о заявках, заказах и т.д. На вход, данный процесс получает статистическую информацию, сохраняемую на первых трёх этапах и критерии формирования

отчётов, а на выходе имеет непосредственно сгенерированную отчётную документацию.

Логически, очевидным является тот факт, что процесс «Обработка заявки» является наиболее функционально нагруженным, обобщает в себе большую часть операций, выполняемых офис-менеджером в рамках организации бизнес-процесса, направленного на предоставление корпоративного транспорта сотрудникам компании. Исходя из этого, была произведена его декомпозиция, представленная на рисунке 1.6.

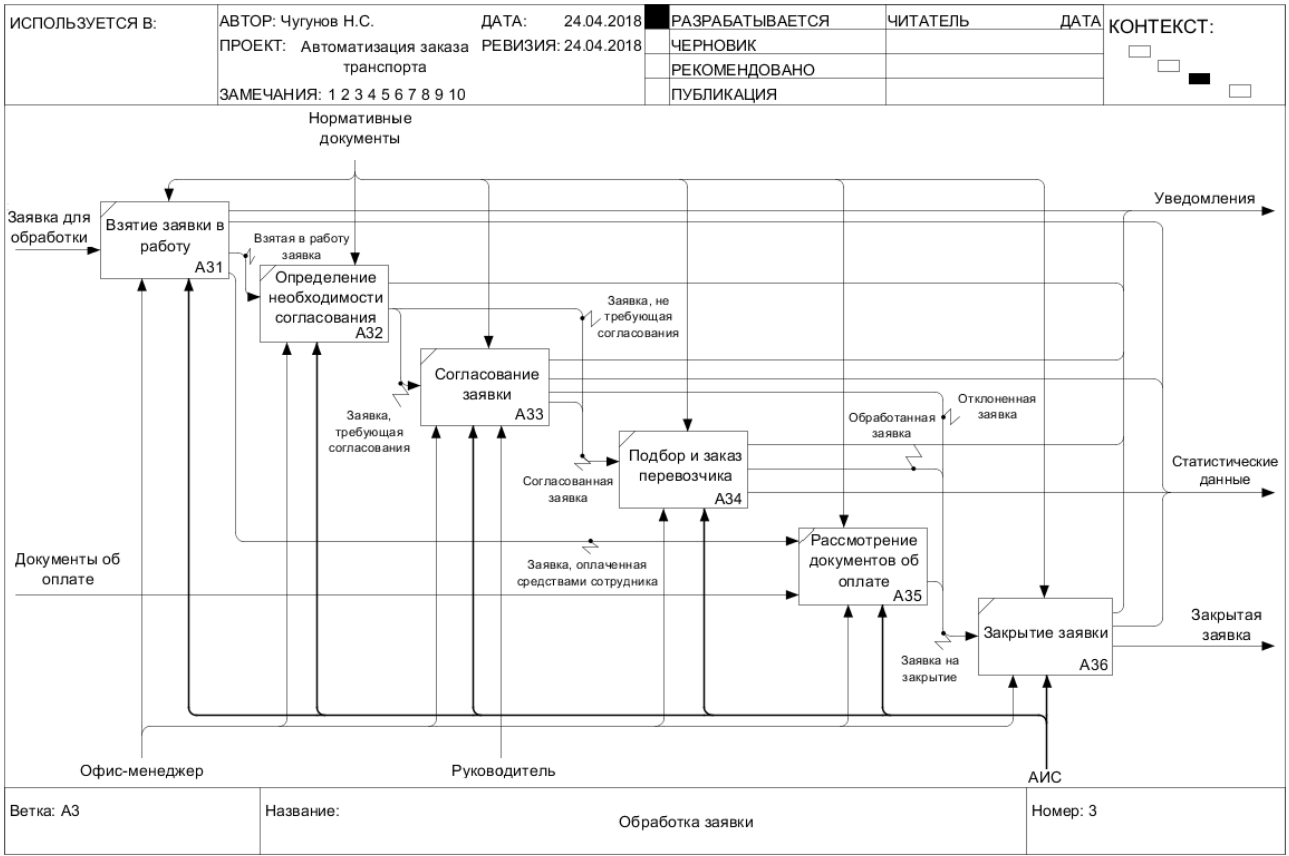


Рисунок 1.6 – Декомпозиция функционального блока «Обработка заявки»

Функциональный блок «Обработать заявку» представлен шестью блоками, описывающими жизненный цикл процесса обработки заявки:

1. Процесс A31 – «Взятие заявки в работу» - подразумевает начало выполнения действий по выбранной заявке для обработки, приходящей на вход. На выходе из данного процесса получается взятая в работу заявка, часть

данных, сохраняемых в качестве статистики, а также соответствующее уведомление, отправляемое сотруднику.

2. Процесс A32 – «Определение необходимости согласования» - подразумевает определение офис-менеджером или другим ответственным лицом необходимости согласования взятой в работу заявки, приходящей на вход, с руководством. На выходе, процесс имеет либо заявку, требующую согласования, которая в дальнейшем рассматривается руководством, либо заявку, не требующую согласования, по которой, непосредственно, подбирается и заказывается транспорт.

3. Процесс A33 – «Согласование заявки» подразумевает передачу заявки на рассмотрение непосредственному руководителю. На выходе из данного процесса получается либо отклоненная заявка, которая подлежит закрытию, либо согласованная заявка, которая продолжает свою дальнейшую обработку. Часть данных сохраняется в качестве статистики, а сотруднику и офис-менеджеру высылаются соответствующие уведомления.

4. Процесс A34 – «Подбор и заказ перевозчика» подразумевает поиск при помощи информационного обеспечения системы возможных вариантов заказа транспорта, удовлетворяющего потребностям сотрудника, а затем связь с таксомоторной компанией и непосредственное оформление заказа.

5. Процесс A35 – «Рассмотрение документов об оплате» подразумевает рассмотрение документов, подтверждающих оплату заявки сотрудником из собственных средств, которые возможно компенсировать.

6. Процесс A36 – «Закрытие заявки» подразумевает закрытие заявки, сохранение части данных как статистики, и отправку уведомлений всем заинтересованным лицам.

Как можно видеть на представленных диаграммах, проектируемая информационная система призвана не кардинально поменять процесс, а лишь делегировать ей часть повторяющейся утомительной рутинной работы, значительно упрощая сокращая время на обработку заявок сотрудников на заказ транспорта.

Выводы по главе 1

В первой главе были рассмотрены основные организационные принципы, на которых в настоящие дни базируется бó льшая часть организаций, описан типовой существующий и предлагаемый рабочий поток бизнес-процессов, направленных на сбор, обработку и непосредственное предоставление корпоративного транспорта сотрудникам различных компаний.

Как было определено, далеко не во всех компаниях процесс заказа транспорта для сотрудников является автоматизированным - учёт и отчетная документация ведется посредством стандартного набора офисных приложений, процесс является достаточно трудозатратным.

Автоматизированная информационная система призвана унифицировать и упростить способ подачи заявок на предоставление корпоративного транспорта сотрудниками различных компаний, предоставить механизмы, позволяющие организовать более высокий уровень оперативности и производительности труда, показателей качества обработки информации, уровня её защищённости, а также сокращения количества бумажной документации.

Глава 2 Проектирование информационной системы

2.1 Объектная модель АИС и её описание

Объектная модель системы – это модель, отражающая её функциональный состав в виде описательного структурированного графического представления. В качестве такого представления может служить диаграмма прецедентов (вариантов использования), позволяющая достаточно полно отразить совокупность актёров и вариантов использования, обозначить отношения между ними, а также, зафиксировать общие границы и контекст моделируемой предметной области на ранних этапах проектирования системы.

В рассматриваемой предметной области, были определены следующие актеры:

- сотрудник – лицо, нуждающееся в транспорте и подающее заявки;
- офис-менеджер – лицо, ответственное за обработку заявок;
- руководитель – лицо, ответственное за согласование заявок;

Перед непосредственным построением диаграммы вариантов использования, необходимо представить прецеденты, подлежащие реализации в проектируемой информационной системе. Перечень и краткое описание основных прецедентов представлены в таблице 2.1.

Таблица 2.1 – Краткое описание прецедентов

Название актера	Название прецедента	Краткое описание прецедента
1	2	3
Сотрудник, офис-менеджер	Зарегистрировать заявку	Указать личные данные, а также требования к поездке, после чего, отправить в систему для последующей обработки
Офис-менеджер	Зарегистрировать заявку на сотрудника	Указать личные данные сотрудника, а также его требования к поездке, после чего, отправить в систему для последующей обработки
Сотрудник	Просматривать список заявок	Просматривать список имеющихся личных заявок
Сотрудник	Просмотреть детали заявки	Открыть определенную заявку для просмотра её полной информации
Сотрудник	Скорректировать заявку	Внести корректировки в заявку при наличии каких-либо ошибок
Сотрудник	Отменить заявку	Остановить процесс выполнения заявки
Сотрудник	Отслеживать статус выполнения заявок	Отслеживать текущий этап выполнения отправленных заявок

Окончание таблицы 2.1

1	2	3
Сотрудник, офис-менеджер	Отправить документ об оплате	Отправить скан-копию чека об оплате для возмещения средств
Офис-менеджер	Просматривать очередь заявок	Просматривать список заявок, ожидающих обработки
Офис-менеджер	Обработать заявку	Подобрать и заказать транспорт, указать детали предстоящей поездки
Офис-менеджер	Рассмотреть документ об оплате	Рассмотреть приложенный к заявке документ об оплате, вынести по нему решение
Офис-менеджер, руководитель	Сформировать отчётность	Сформировать отчётную документацию по заданным критериям
Офис-менеджер	Заказать транспорт и указать детали поездки	При помощи информационного обеспечения АИС, подобрать перевозчика, связаться с ним, заказать транспорт и указать детали в заявке
Руководитель	Просматривать список заявок на согласование	Просматривать список заявок, ожидающих рассмотрения
Руководитель	Рассмотреть заявку	Рассмотреть детали заявки и вынести по ней решение

Представленные в таблице актеры и варианты использования, а также их взаимосвязи отражены на рисунке 2.1.

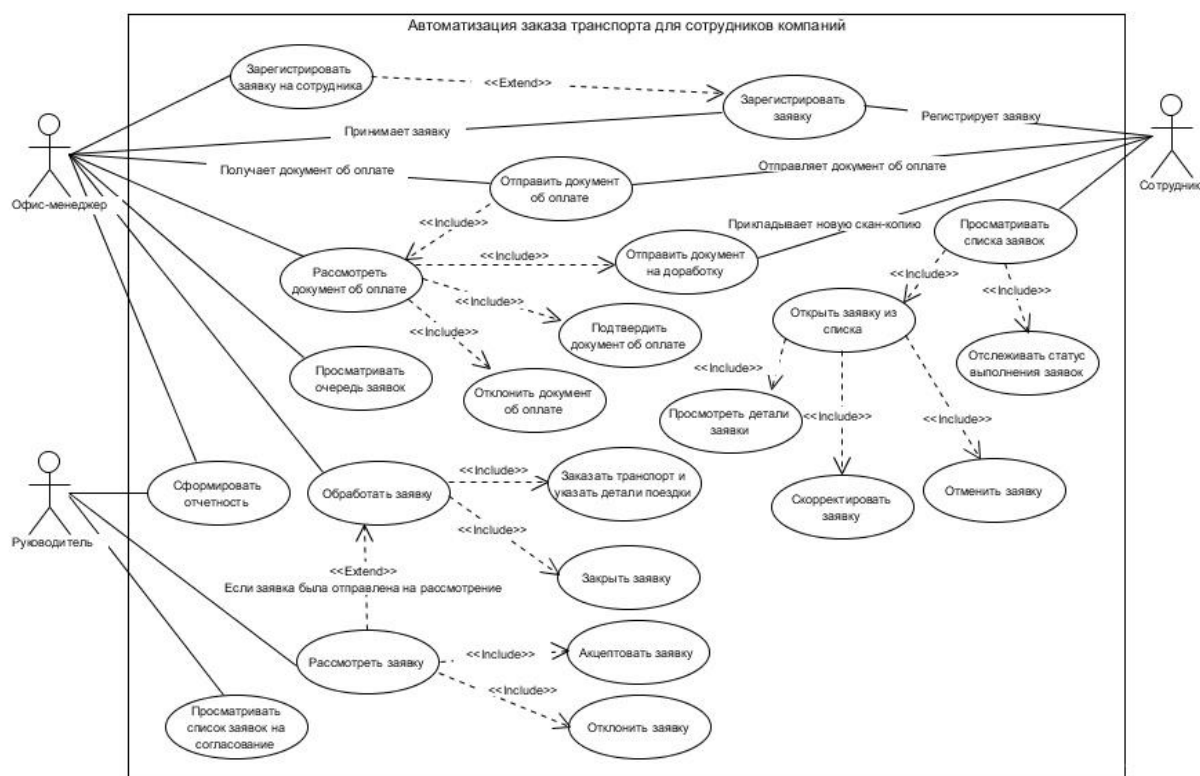


Рисунок 2.1 – Диаграмма вариантов использования

Построенная диаграмма вариантов использования отражает основные аспекты разработки, отображает границы моделируемой предметной области, взаимоотношения задействованных в ней актеров, а также зависимости ранее обозначенных прецедентов, подлежащих реализации в качестве функциональности, проектируемой автоматизированной информационной системы друг от друга.

В дополнение к построенной диаграмме, описывающей возможные прецеденты определенных ранее актеров, для описания объектного состава рассматриваемой предметной области и обозначения их зависимостей друг от друга, необходимо построить доменную модель её объектов. Доменный объект представляет собой сущность из модели предметной области, имеющую непосредственное отношение к информационной системе и реализующую её бизнес-логику [25]. Представленная в приложении А диаграмма классов наглядно отражает модель доменных объектов проектируемой информационной системы, их содержания и взаимосвязей.

Фактически, представленная доменная модель представляет собой концептуальное, независящее от реализации описание анализируемой предметной области. Её использование позволяет сфокусироваться на самом «ядре» будущей информационной системы, полностью абстрагироваться от деталей её технической реализации на данном этапе. Несмотря на это, следует обратить внимание на тот факт, что построение и применение данной модели, также, значительно облегчает процесс разработки, позволяя реализовывать архитектурно «чистую», понимаемую и легко расширяемую кодовую базу, предоставляя возможность применения различного рода паттернов проектирования и прочих успешных практик инжиниринга программных продуктов.

2.2 Логическая модель АИС и её описание

Логическая модель АИС подразумевает под собой отражение взаимодействия системы и определенных ранее актёров во времени,

передаваемых между ними сообщений, запросов, возвращаемых на них ответов и, наконец, логики их обработки.

Нагляднее всего, логическую модель рассматриваемой информационной системы можно представить в виде набора диаграмм последовательностей, прямое предназначение которых – уточнение диаграммы прецедентов посредством отображения общего сценария взаимодействия и особенностей обмена сигналами, сообщениями или пакетами данных между системой и непосредственно задействованными в её работе актерами.

Диаграмма, описывающая стандартный процесс оформления и обработки заявки представлена на рисунке 2.2.

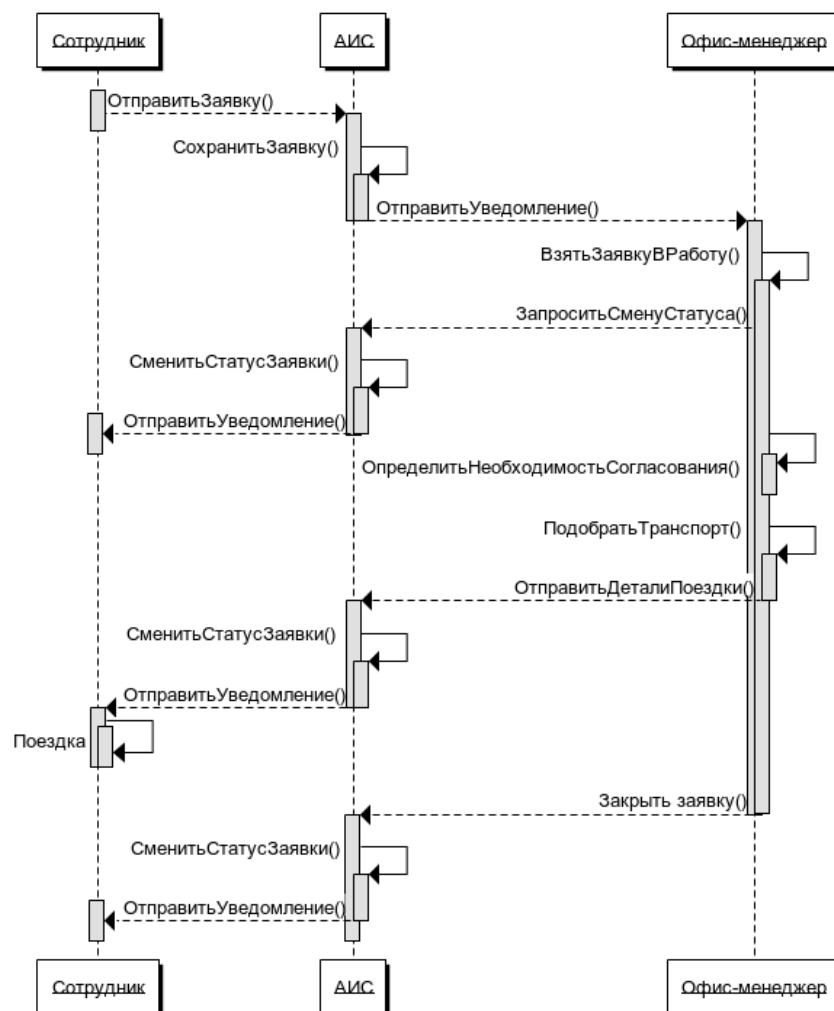


Рисунок 2.2 – Диаграмма последовательности обобщенного процесса подачи и обработки заявок на заказ транспорта

Взаимодействие между субъектами и объектами процесса осуществляется по следующему сценарию:

1. Сотрудник формирует и отправляет требования к поездке;
2. Система регистрирует новую заявку и отправляет офис-менеджеру соответствующее уведомление.
3. Офис-менеджер берет заявку в работу, инициирует смену её статуса, о чем сотрудник, её оформивший, получает соответствующее уведомление.
4. Офис-менеджер, используя информационное обеспечение АИС подбирает таксомоторную компанию, связывается с ней, заказывает транспорт, после чего, указывает детали предстоящей поездки в заявке и сохраняет в системе, инициируя смену ее статуса, обозначая её как успешно выполненную. Сотрудник, оформивший заявку получает соответствующее уведомление.
5. Офис-менеджер закрывает заявку. Сотрудник, оформивший заявку получает соответствующее уведомление.

Как уже говорилось ранее, при обработке заявки офис-менеджером, может возникнуть необходимость в согласовании её с руководителем, в результате которого может быть получено как положительное, так и отрицательное решение. Диаграмма последовательности, описывающая логику отправки заявки, требующей согласования на рассмотрение руководителю, её непосредственного рассмотрения, а также обработки полученных результатов согласования заявки, представлена на рисунке 2.3.

Взаимодействие между субъектами в данном случае осуществляется по следующему сценарию:

1. Офис-менеджер определил необходимость согласования заявки с руководителем. Он выбирает руководителя, которому необходимо передать заявку на согласование, после чего изменяет ее статус. Выбранный руководитель получает уведомление о получении новой заявки на согласование, а сотрудник, её оформивший, получает уведомление о передаче заявки на согласование.

2. Руководитель согласует заявку и сохраняет решение по ней в системе, инициируя смену её статуса, после чего заявка отправляется офис-менеджеру на дальнейшую обработку. Оформивший заявку сотрудник и офис-менеджер получают уведомления решения.

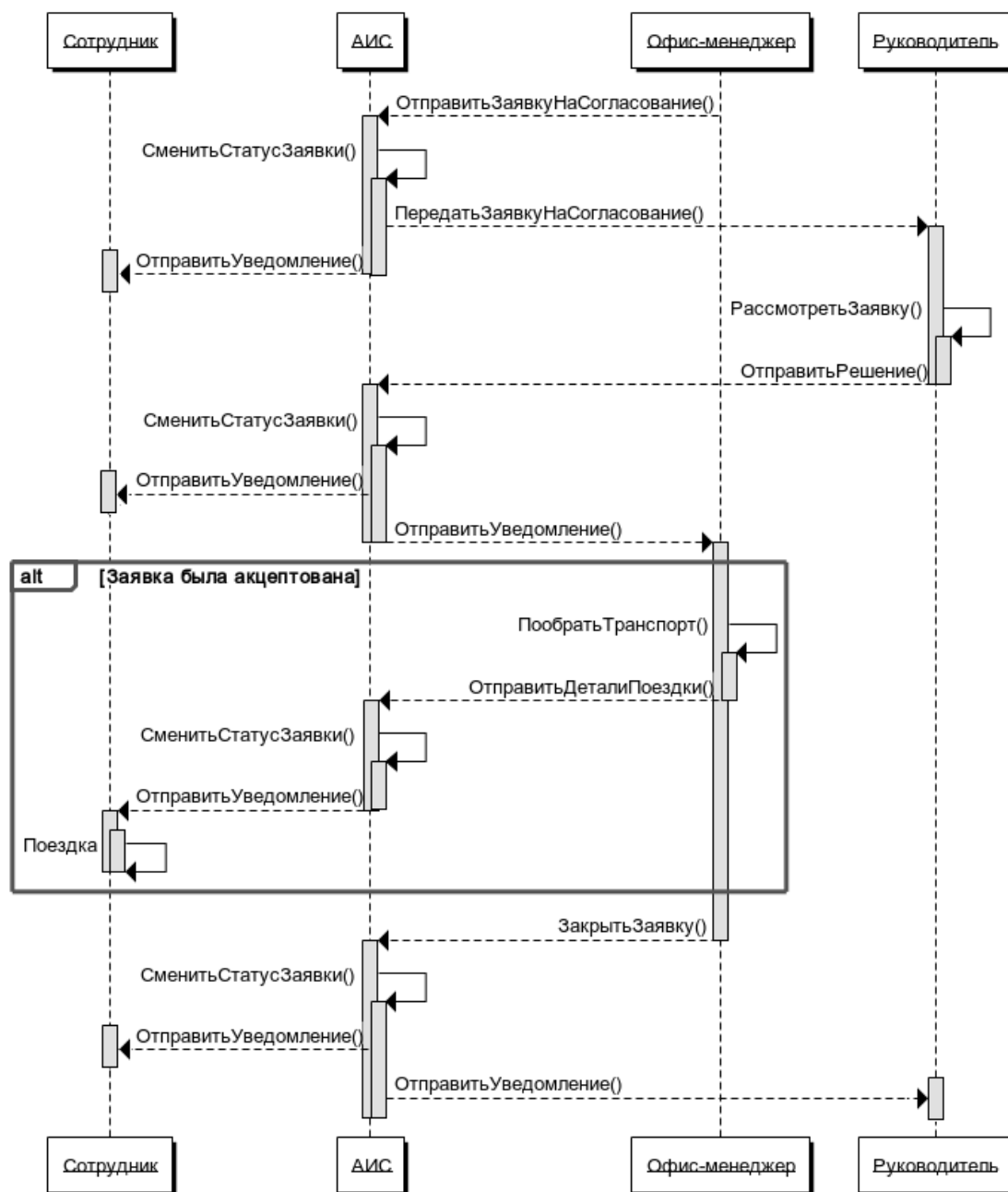


Рисунок 2.3 – Диаграмма последовательности обработки заявки после отправки на согласование

3. В том случае, если решение было положительным, заявка передается офис-менеджеру на дальнейшую обработку.

3.1 Офис-менеджер, используя информационное обеспечение АИС подбирает таксомоторную компанию, заказывает транспорт, после чего, указывает детали в заявке и сохраняет в системе, иницируя смену ее статуса.

3.2 Сотрудник, оформивший заявку получает соответствующее уведомление.

3.3 При наличии успешной поездки, сотрудник, оформивший заявку обозначает её как успешно выполненную, о чём офис-менеджер получает соответствующее уведомление.

3.4 Офис-менеджер закрывает заявку. Оформивший заявку сотрудник получает соответствующее уведомление.

4. В том случае, если решение было отрицательным, офис-менеджер закрывает заявку. Руководитель и сотрудник получают соответствующие уведомления.

Отдельным пунктом выделен процесс отправки и рассмотрения документа, подтверждающего оплату сотрудником транспорта из личных средств, для дальнейшей их компенсации. Диаграмма последовательности данного процесса представлена на рисунке 2.4.

Взаимодействие между субъектами и объектами процесса осуществляется по следующему сценарию:

1. После поездки, в том случае, если оплата совершалась не по корпоративной карте компании, а из личных средств сотрудника и ему необходима компенсация транспортных расходов, он отправляет чек об оплате;

2. Офис-менеджер получает уведомление о поступлении документа об оплате на рассмотрение;

3. Офис-менеджер рассматривает приложенный документ и выносит по нему положительное или отрицательное решение, либо отправляет на доработку, о чем сотрудник получает соответствующее уведомление;

3.1 В том случае, если документы об оплате отправлены на доработку, сотрудник исправляет недочеты и отправляет документы заново.

3.2 Процесс продолжается до тех пор, пока документы не будут подтверждены или отклонены.

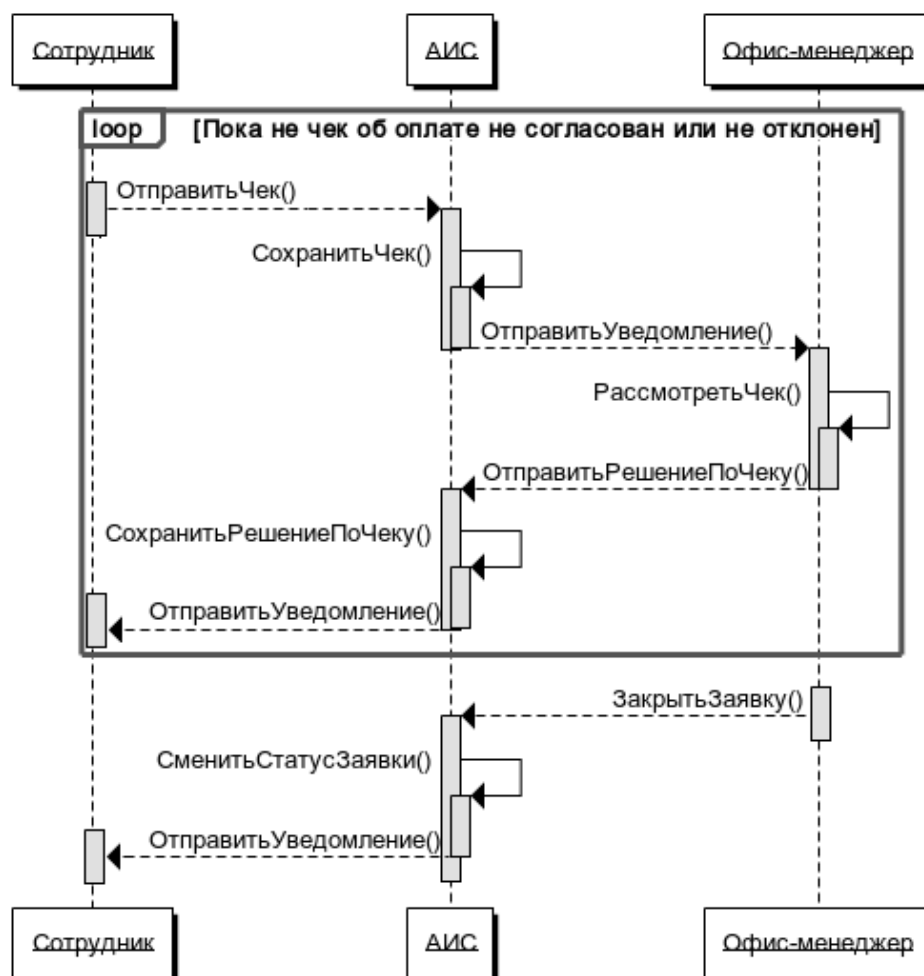


Рисунок 2.4 – Диаграмма последовательности процесса отправки и рассмотрения документа об оплате

4. Офис-менеджер закрывает заявку, о чем сотрудник получает соответствующее уведомление.

Таким образом, представленные на рисунках диаграммы последовательностей описывают наиболее значимые блоки рабочего потока её функционирования, направленные на организацию процесса заказа транспорта для сотрудников компании, наглядно отражают последовательность и условия их выполнения.

2.3 Функциональная модель АИС и её описание

Автоматизированная информационная система, разрабатываемая с целью автоматизации процесса заказа транспорта для сотрудников компаний, должна отвечать функциональным требованиям, определенным на более ранних этапах её проектирования.

В п. 2.1 были представлены объектная модель системы, определены три актёра. Соответственно, функциональность системы должна распределяться между тремя основными ролями – сотрудник, офис-менеджер и руководитель.

На рисунке 2.5 представлена функциональная модель информационной системы, отражающая основные функциональные возможности каждого из актеров.



Рисунок 2.5 – Функциональная модель информационной системы

Пользователь системы, имеющий роль «Сотрудник» имеет возможность просматривать список личных заявок, регистрировать новые или отслеживать статус выполнения заявок, находящихся в работе, а также корректировать или отменять их в тех случаях, когда это необходимо. Помимо этого, он может прикладывать документы об оплате поездки в тех случаях, когда это необходимо и получать различного рода уведомления, отправляемые ему системой, например, при наличии каких-либо изменений в содержании или статусе оформляемых им заявок.

Пользователь системы, имеющий роль «Офис-менеджер» имеет доступ к функциональности, позволяющей регистрировать новые заявки на сотрудников без их непосредственного участия, отслеживать наличие в общем списке новых, подлежащих к обработке заявок, а также просматривать общую историю. Помимо этого, он имеет доступ к функциональности, позволяющей обработать заявку, т.е. с использованием информационного обеспечения АИС подобрать транспорт, полностью удовлетворяющий требованиям сотрудника, а также определить контакты таксомоторной компании, его предоставляющей, для связи с ней, совершения заказа и последующей отправки этой информации сотруднику. Наконец, офис-менеджер может рассматривать документы об оплате, прикладываемые сотрудником к заявке в случае необходимости, и формировать отчетную документацию.

Наконец, пользователь, имеющий роль «Руководитель» имеет возможность отслеживания список заявок, требующих согласования, акцептации или отклонения заявок, а также формирования отчетной документации при наличии такой необходимости.

На данном этапе, построение моделей, непосредственно описывающих разрабатываемую информационную систему с точки зрения функциональности, актеров, участвующих в ней, а также процесса их взаимодействия окончено. Теперь, необходимо представить информационное обеспечение системы – используемые документы, справочники и классификаторы, а также возможные документы, получаемые в качестве результатов её функционирования, после чего, используя данную информацию, спроектировать будущее информационное хранилище её данных.

2.4 Информационное обеспечение АИС

2.4.1 Описание используемых нормативно-справочных документов

В контексте автоматизированных систем, документы нормативно-справочной информации представляют собой некие информационные ресурсы компаний, их использующих, которые являются основными для унификации и

нормализации данных, сопровождают протекающие внутри организаций бизнес-процессы [24].

Рассматриваемая автоматизированная информационная система в своей работе опирается на ряд документов, являющихся нормативно-справочными. Одним из таких справочников, определяющим перечень адресов точек интереса компании может служить документ «Адресная книга», вид и пример заполнения которого представлен на рисунке 2.6.

ООО «Компания»		
Адресная книга от «__» _____ 20__ г.		
№ п/п	Наименование точки интереса	Адрес
1	2	3
1	Офис ООО «Компания» в г. Тольятти	44500, г. Тольятти, ул. Фрунзе, д. 32
2	Банк «Компаньон» в г. Тольятти	445048, г. Тольятти, ул. Фрунзе, д. 35
Подпись директора _____		

Рисунок 2.6 – Вид справочника «Адресная книга»

Следующим нормативно-справочным документом, описывающим иерархию компании, является документ «Организационная структура компании», упрощенный вид и пример заполнения которого представлен на рисунке 2.7.

ООО «Компания»		
Организационная структура компании от «__» _____ 20__ г.		
Код	Наименование подразделения	Возглавляющее подразделение
1	2	3
1	ООО «Компания»	-
2	Административный отдел ООО «Компания»	ООО «Компания»
3	Отдел разработки ООО «Компания»	ООО «Компания»
4	Подразделение Java-разработки ООО «Компания»	Отдел разработки ООО «Компания»
5	Подразделение .NET-разработки ООО «Компания»	Отдел разработки ООО «Компания»
Подпись директора _____		

Рисунок 2.7 – Вид документа «Организационная структура компании»

Наконец, последним справочным документом, предоставляющим информацию об организационном устройстве компании, является документ «Перечень сотрудников компании». В контексте рассматриваемой предметной области, данный документ является справочным, поскольку лишь предоставляет информацию о сотрудниках, числящихся в штате компании, проектируемая информационная система лишь использует данные сведения, не оперируя ими напрямую. Упрощенный вид и пример заполнения данного документа представлен на рисунке 2.8.

ООО «Компания»				
Штат компании от «__» _____ 20__ г.				
Таб.номер	ФИО сотрудника	Номер телефона	Подразделение	Должность
1	2	3	4	5
101	Иванова Любовь Владимировна	+7 927 463-37-46	Администрация	Директор
102	Ветров Владимир Сергеевич	+7 927 463-37-56	Отдел разработки	Начальник отдела
103	Сидоров Игорь Степанович	+7 987 546-73-12	Подразделение Java-разработки	Разработчик
104	Ольхов Василий Владленович	+7 976 377-63-56	Подразделение .NET-разработки	Разработчик
105	Кондратьев Глеб Викторович	+7 968 239-48-34	Администрация	Офис-менеджер
Подпись директора _____				

Рисунок 2.8 – Вид документа «Перечень сотрудников компании»

Поскольку, в рамках данной работы рассматривается процесс предоставления корпоративного транспорта сотрудникам различных компаний, еще одним справочным документом является «Перечень обслуживающих компаний», описывающий список подрядчиков-таксомоторных компаний, с которыми сотрудничает организация. Вид и пример заполнения данного документа представлен на рисунке 2.9.

ООО «Компания»				
Перечень обслуживающих таксомоторных компаний				
от «__» _____ 20__ г.				
№ п/п	Наименование компании	Предоставляемые классы	Контактный телефон	Номер корпоративной карты
1	2	3	4	5
1	Такси «Везет»	А, В	8 (8482) 52-23-23	19238456106
2	Такси «Амбассадор»	В, С, Е	8 (8482) 21-45-62	1K215122562
3	Такси «Командирь»	В, Е, J	8 (8482) 41-35-29	94726361636
Подпись директора _____				

Рисунок 2.9 – Вид документа «Список обслуживающих компаний»

Следует отметить, что документы нормативно-справочной информации могут быть как динамически изменяемыми, так и статическими. Одним из таких справочников, необходимым для обеспечения возможности ранжирования заявок в общем списке, а также обозначения критичности и важности их выполнения, является перечень приоритетов выполнения заявки, вид которого представлен в таблице 2.2.

Таблица 2.2 – Перечень приоритетов выполнения заявки

Код приоритета	Название приоритета
1	2
1	Наивысший приоритет
2	Высокий приоритет
3	Средний приоритет
4	Низкий приоритет

Отдельно, как частный тип документов нормативно-справочной информации, следует отметить классификаторы различного вида, назначения и содержания, используемые практически каждой информационной системой. И, поскольку рассматриваемая информационная система предполагает оперирование большим количеством адресной информации, было принято решение об использовании классификатора адресов Российской Федерации (далее – КЛАДР). Классификации подлежат отдельные элементы адресов, именуемые адресными объектами. Классификатор построен в виде иерархии, которая представляется шестью уровнями:

1) первый уровень классификации включает в себя объекты Федерального значения, входящие в состав РФ – области, края, автономные области, города Москва, Санкт-Петербург и Байконур и т.д;

2) второй уровень классификации включает в себя районы, относящиеся к адресным объектам первого уровня;

3) третий уровень классификации включает в себя города и поселки городского типа регионального и районного подчинения;

4) четвертый уровень классификации включает города и поселки городского типа, подчиненные администрациям объектов третьего уровня иерархии;

5) пятый уровень классификации включает улицы населенных пунктов третьего и четвертого уровней;

6) шестой уровень классификации включает дома улиц городов.

В общем представлении, структуру кодового обозначения адреса можно представить в следующем виде:

СС RPP ГГГ ППП УУУУ КК,

где: СС – код субъекта Российской Федерации (региона);

RPP – код района;

ГГГ – код города;

ППП – код населенного пункта;

УУУУ – код улицы;

КК – код актуальности наименования.

Так, например, город Тольятти будет представлен кодом 63000000700000, а код улицы 70 Лет Октября кодом 630000007000007300.

Помимо адресной информации, проектируемая информационная система предполагает оперирование статической справочной информацией, описывающей классы автомобилей. В целях обеспечения непротиворечивости и актуальности хранимой информации, было принято решение об использовании классификатора классов автомобилей, определенный Европейской

экономической комиссией. В списке имеется 9 классов, их перечень представлен в таблице 2.3.

Таблица 2.3 – Классификатор классов автомобилей

Код класса	Название класса
1	2
1	A: Mini cars (микроавтомобили)
2	B: Small cars (малые автомобили)
3	C: Medium cars (европейский «средний класс»)
4	D: Larger cars (большие семейные автомобили)
5	E: Executive cars («бизнес-класс»)
6	F: Luxury cars (представительские автомобили)
7	S: Sport coupes (спорткупе)
8	M: Multipurpose cars (минивэны и УПВ)
9	J: Sports utility (SUV)

Резюмируя, можно сказать, что использование справочников, классификаторов и систем кодирования призвано унифицировать способы хранения и обработки информации АИС, сократить объемы передаваемой и хранимой адресной информации, поддерживать её актуальность, достоверность и корректность.

2.4.2 Характеристика входной и оперативной информации

Проектируемая автоматизированная информационная система направлена на автоматизацию процесса предоставления сотрудникам различных компаний корпоративного транспорта. Соответственно, основным входным и оперативным документом является непосредственно электронная заявка на заказ транспорта. Вид данного документа представлен на рисунке 2.10.

ООО «Компания»	
Заявка на заказ транспорта № ____ от «__» _____ 20__ г.	
Фамилия, имя, отчество сотрудника: _____	
Подразделение: _____	
Предполагаемая дата поездки: «__» _____ 20__ г.	
Необходимый класс автомобиля: _____	
Адрес пункта отправки: _____	
Адрес пункта назначения: _____	
Контактный номер телефона: +7 (____) ____-____-____	
Контактный e-mail: _____@_____	
Наличие багажа: _____	
Подпись сотрудника: _____	
Подпись офис-менеджера: _____	
Дата согласования: «__» _____ 20__ г.	
Подпись руководителя: _____	

Рисунок 2.10 – Вид входного документа «Заявка на заказ транспорта»

Как можно видеть на рисунке, документ содержит все необходимые и достаточные данные для осуществления заказа автотранспорта в таксомоторных компаниях: данные сотрудника, которому необходим транспорт, требуемый класс заказываемого транспорта, предполагаемые дата и время отправки, пункт отправки и прибытия, а также контактная информация, с помощью которой, с сотрудником можно будет связаться и сообщить ему детали предстоящей поездки.

После заполнения данной формы сотрудником, она передается офис-менеджеру или другому ответственному лицу, а затем, берется в работу; по требованиям, указанным в заявке, подбирается перевозчик, при необходимости, заявка согласовывается с непосредственным руководителем сотрудника. Результаты обработки заявки сообщаются сотруднику лично.

Еще одним документом, который в рассматриваемом контексте может определяться, как источник оперативной информации, является документ «Квитанция на оплату пользования такси». Фактически, это кассовый чек или квитанция, выдаваемая перевозчиком после совершения поездки, скан-копия которой может прикладываться сотрудником к заявке в тех случаях, когда по тем или иным причинам он был вынужден совершить оплату поездки из

личных средств, и компания готова предоставить финансовую компенсации затраченных средств после рассмотрения и подтверждения приложенного документа сотрудником, ответственным за обработку заявок. Вид данного документа представлен на рисунке 2.11.

Чек	

Наименование перевозчика _____	
ИНН _____	

№ Чека: _____	№ Автомобиля _____
№ Смены: _____	Дата поездки: «__» ____ 20__ г.

Наименование услуги: _____	
Цена за единицу пред. услуги:	_____ р.
Количество:	_____
ИТОГО:	_____ р.

Рисунок 2.11 – Вид документа «Квитанция на оплату пользования такси»

Представив и описав основные документы-источники справочной и оперативной информации, необходимо перейти к непосредственному описанию документов, получаемых на выходе в качестве результатов работы системы.

2.4.3 Характеристика выходной информации

Практически каждая информационная система, как результат своей работы, предоставляет выходные документы - как правило, это отчетная документация или какие-либо статистические данные. С учётом текущих источников входной, оперативной и справочной информации, а также поставленных перед разработкой задач, представляющими наибольший возможный интерес были определены следующие документы:

1. Документ «Отчёт о поездках сотрудника за период», представленный на рисунке 2.12 - данный документ наглядно отображает информацию по поездкам, совершенным определенным сотрудником за указанный период времени.

ООО «Компания» Отчёт № «__» от «__» _____ 20__ г. о поездках сотрудника _____ с _____ по _____ 20__ г.				
№ заявки	Дата поездки	Пункт отправки	Пункт назначения	Перевозчик
1	2	3	4	5
Общее количество заявок:				
Подпись офис-менеджера: _____				
Подпись директора: _____				

Рисунок 2.12 – Вид документа «Отчёт о поездках сотрудника за период»

2. Документ «Отчёт о поездках сотрудников подразделения за период», обобщающий и отражающий общую историю поездок, совершенных сотрудниками заданного подразделения за указанный период. Вид документа представлен на рисунке 2.13.

ООО «Компания» Отчёт № «__» от «__» _____ 20__ г. о поездках сотрудников подразделения _____ с «__» _____ 20__ г. по «__» _____ 20__ г.					
№ п/п	ФИО сотрудника	Дата поездки	Пункт отправки	Пункт назначения	Перевозчик
1	2	3	4	5	6
Общее количество заявок:					
Подпись директора: _____					

Рисунок 2.13 – Вид документа «Отчет о поездках сотрудников подразделения»

3. Документ «Список заказов, совершенных по заданной корпоративной карте за указанный период», представленный на рисунке 2.14. Данный документ может быть полезен для сверки заявок, зарегистрированных в системе с выставленным таксомоторной компанией счётом.

ООО «Компания» Отчёт № «__» от «__» _____ 20__ г. о поездках сотрудников предприятия по карте № _____ с «__» _____ 20__ г. по «__» _____ 20__ г.						
№ п/п	ФИО сотрудника	Подразделение	Дата поездки	Пункт отправки	Пункт назначения	Класс автомобиля
1	2	3	4	5	6	7
Общее количество заявок по карте за период:						
Подпись офис-менеджера _____						
Подпись директора _____						

Рисунок 2.14 – Вид документа «Отчёт о поездках сотрудников предприятия за указанный период»

4. Документ «Отчёт о транспортных расходах сотрудника», представленный на рисунке 2.15 - данный документ отражает список поездок заданного сотрудника за указанный период, которые были оплачены им из собственных средств, и, соответственно, по которым необходимо осуществить компенсацию затраченных средств.

ООО «Компания» Отчёт № «__» от «__» _____ 20__ г. о транспортных расходах сотрудника _____ с «__» _____ 20__ г. по «__» _____ 20__ г.						
№ заявки	Дата поездки	Пункт отправки	Пункт назначения	Перевозчик	Стоимость поездки	Номер чека
1	2	3	4	5	6	7
Общая сумма компенсации:						
Подпись офис-менеджера: _____						
Подпись директора: _____						

Рисунок 2.15 – Вид документа «Отчёт о транспортных расходах сотрудника за период»

Поскольку, данный отчет является агрегирующим, он также отражает общую сумму средств, затраченную за указанный период, подлежащую

возврату сотруднику. Общая сумма затрат за период вычисляется по представленной формуле 2.1:

$$\text{Общая сумма затрат} = \sum_{i=1}^n \text{Стоимость поездки}_i \quad (2.1)$$

На данном этапе, процесс описания информационного обеспечения разрабатываемой информационной системы можно считать завершённым. Далее, необходимо перейти к непосредственному построению логической модели хранилища её данных, с учётом необходимости соблюдения целостности, непротиворечивости, консистентности и защищенности используемой информации.

Выводы по параграфу 2.4

В параграфе 2.4 было представлены и классификаторы, справочники и документы, представляющие собой источники входной и оперативной информации, используемые системой, а также документы, получаемые в качестве результатов её работы. Поскольку, данная информация является основополагающей для определения состава данных, следующим этапом разработки проектного решения является непосредственное построение и описание логической модели данных системы, являющейся прототипом физической реализации будущего информационного хранилища.

2.5 Логическая модель данных АИС и её описание

Логическая модель данных служит для описания понятий рассматриваемой предметной области, полностью абстрагированного от среды и способа их хранения. Также, данная модель описывает взаимосвязи данных понятий и определяет ограничения, накладываемые на данные предметной областью. Логическая модель данных представляет собой начальный прототип физической реализации их хранилища.

При построении логической модели хранилища данных информационной системы, необходимо также брать в учет требования, определяемые правилами

нормализации данных. Это важно, поскольку позволяет значительно оптимизировать способы хранения и обработки данных путем сокращения возможного числа противоречивости, избыточности и неконсистентности информации, хранимой в базе данных.

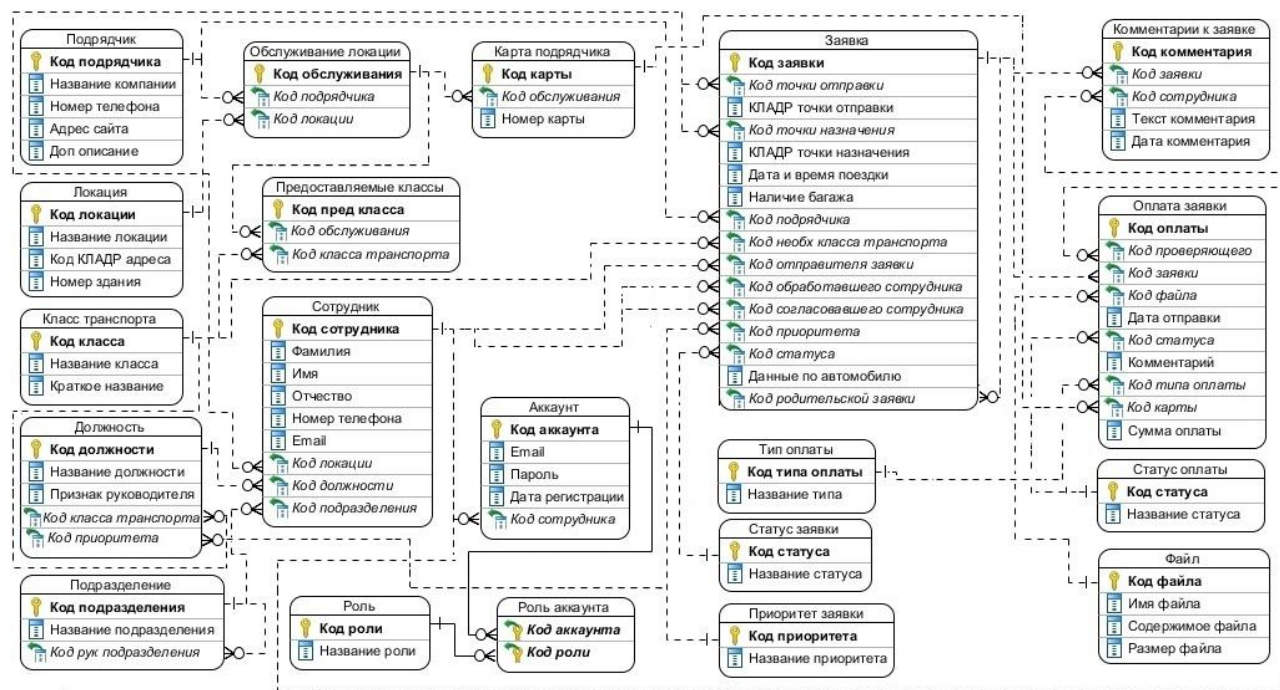


Рисунок 2.16 – Логическая модель хранилища данных АИС

На рисунке 2.16 представлена логическая модель данных автоматизированной информационной системы, отражающая набор используемых сущностей с их атрибутами, ключами, а также взаимосвязями.

Построенная модель наглядно определяет, описывает реальные информационные объекты рассматриваемой предметной области и служит отправной точкой для физической реализации хранилища данных информационной системы.

Выводы по главе 2

В главе 2, посвященной разработке проекта рассматриваемой информационной системы, были представлены модели, наглядно отражающие функциональный, объектный и информационный состав будущего программного продукта.

Помимо этого, с учетом объема и специфики документов, определяющих информационное обеспечение системы и результаты её работы, а также требований, определяемых правилами нормализации, была разработана и представлена логическая модель будущего физического хранилища данных.

На данном этапе, процесс описания формальной спецификации и разработки проекта решения можно считать завершенным. Следующим этапом решения задачи является определение архитектурного подхода и технологического стека, используемого для реализации, а также непосредственная техническая реализация её информационного хранилища и программных компонентов в соответствии с предъявляемым к ним требованиям и, наконец, описанием реализованной функциональности.

Глава 3 Техническая реализация компонентов АИС

3.1 Описание архитектуры АИС

В сфере информационных технологий, под термином «архитектура» принято подразумевать концепцию, определяющую её структурную модель, распределение реализуемой функциональности между составляющими её компонентами, взаимосвязь и зависимость данных компонентов друг от друга. В настоящее время, существует достаточно большое количество различных архитектурных решений, являющихся действенными и актуальными в зависимости от целей, особенности и нужд разработки, однако, наибольшую популярность получила многозвенная архитектура «клиент-сервер», предполагающая распределение сетевой и вычислительной нагрузки между поставщиками услуг, называемых серверами и их потребителями, называемых клиентами. Данный архитектурный подход предполагает разделение информационной системы на три основных уровня (слоя):

1. Клиентское приложение (слой представления) – интерфейсный, графический компонент системы, поставляемый конечным пользователям системы. Исходя из требований безопасности, масштабируемости и надежности, данный компонент не должен иметь прямого доступа к базе данных, быть нагружен бизнес-логикой или хранить её состояние.

2. Сервер приложений (слой бизнес-логики) – представляет собой среднее звено архитектуры, сосредотачивающим на себе всю логику бизнес-правил информационной системы.

3. Сервер базы данных – компонент системы, обеспечивающий хранение данных, которыми оперирует система, а также предоставление этих данных запрашивающим их компонентам. Является самым нижним звеном архитектуры, напрямую может вызываться только сервером приложений.

Модель многозвенной клиент-серверной архитектуры систем web-среды представлена на рисунке 3.1, наглядно демонстрируя схему взаимодействия её компонентов.

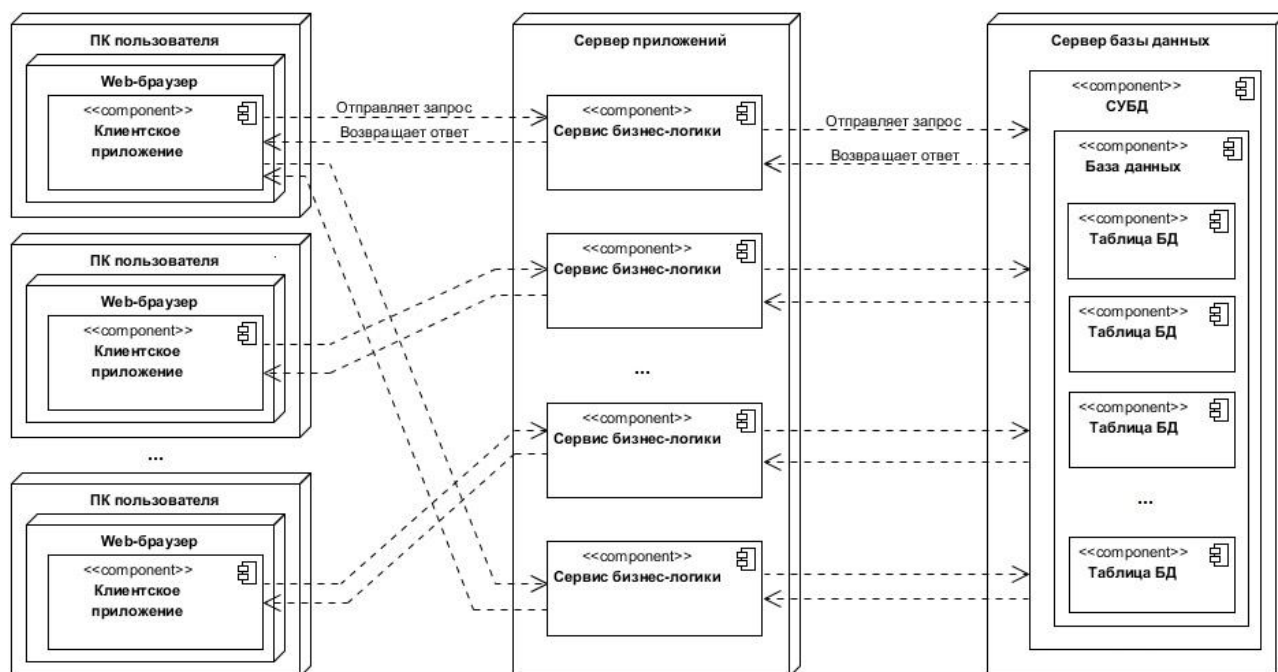


Рисунок 3.1 – Модель многозвенной клиент-серверной архитектуры

В информационных системах, организованных по данному принципу, «тонким» клиентом является пользовательский интерфейс системы, представленный в виде веб-страниц. Данный компонент реализует простейшие операции, такие как валидация вводимых пользователем данных или прочие манипуляции с графическими элементами. Фактически, клиентское приложение представляет собой самое верхнее звено архитектуры, непосредственно не используя не одним из других её слоев, основное его назначение - осуществление необходимых запросов к серверу приложений, получение и обработка ответов, и отображение их пользователю.

Сервер приложений в данном архитектурном решении, является «толстым», поскольку несет на себе всю основную вычислительную логику системы, являясь промежуточным звеном между клиентом и сервером базы данных. Он должен быть спроектирован таким образом, чтобы обеспечить при необходимости, его расширение или полную замену с возникновением минимального количества ошибок, а также внесением дополнительных изменений в программный код системы.

Сервер базы данных предоставляет информационное хранилище системы, находящееся под управлением СУБД. Данное звено архитектуры только содержит данные, которыми оперирует система, никакой вычислительной логики на себе не несет, однако позволяет средствами СУБД, (триггеры, хранимые процедуры, функции) реализовывать дополнительную логику и на этой стороне, при возникновении такой необходимости.

Для рассматриваемой информационной системы, было принято решение об использовании именно этой архитектурной модели, поскольку её использование открывает ряд преимуществ:

1. Концентрация основной прикладной логики системы на сервере приложений позволяет предъявлять низкие требования к производительности и техническим характеристикам клиентских терминалов, что, следовательно, обеспечивает их низкую стоимость;

2. Равномерное распределение нагрузки на сеть определяет низкие требования к пропускной способности канала связи между клиентом и сервером приложений;

3. Избавление от дублирования программного кода уровня прикладной логики на клиентских терминалах;

4. Хранение данных на сервере, обеспечивающим гораздо более высокий уровень их защищенности по сравнению с большим количеством клиентов;

5. Простота и удобство конфигурирования системы при проведении плановых работ, а также устранения возникших неисправностей и сбоев на каком-либо из уровней системы.

6. Легкая расширяемость, сопровождение или полная замена любого из компонентов системы.

Определив архитектурное представление будущей информационной системы, следует перейти к непосредственному определению средств технической реализации её функциональных компонентов и хранилища данных, с учётом предъявляемых архитектурным решением требований.

3.2 Технологическое обеспечение решения

3.2.1 Определение набора технологий для реализации серверного программного обеспечения АИС

Выбор технологического стека для разработки программного обеспечения информационной системы – одно из наиболее важнейших решений, принимаемых перед его непосредственной реализацией. Оно должно приниматься обдуманно, обоснованно, и, по возможности с учётом дальнейших перспектив развития - выбранный набор технологий, должен учитывать требования, предъявляемые выбранной архитектурной моделью, позволять решить эти задачи наиболее простым и гибким способом, однако, не исключая возникновения актуальной необходимости её расширения в дальнейшем.

Учитывая, что рассматриваемая информационная система будет иметь многозвенную архитектуру, набор технологий для ее реализации так же, будет разделяться на три группы, с учётом их принадлежности к направлению разработки: frontend-технологии, backend-технологии и система управления базами данных.

Для реализации серверной части рассматриваемой информационной системы, был выбрана технология компании Microsoft – платформа .NET Framework, в частности, язык программирования C#, с помощью которого, в большинстве случаев, ведется разработка под данную платформу. Данная технология была выбрана ввиду наличия следующих, основных отличительных особенностей:

- с одной стороны, язык программирования C# является синтаксически чистым и простым в освоении, а с другой стороны – является мощным инструментом для решения огромного количества технических задач;
- поскольку C# является основным языком разработки приложений под .NET Framework, он позволяет использовать его совместно с другими языками программирования, для которых имеются .NET-компиляторы;

– C# позволяет написание и использование методов расширения, позволяющих добавлять функциональность даже к тем классам, доступ к исходному коду которых отсутствует;

– C# реализует функциональность встроенного языка структурированных запросов LINQ, предоставляющим значительное удобство и широкий функционал по обработке данных, позволяя формировать программные запросы к объектам в SQL-подобном синтаксисе и точечной нотации. LINQ является мощнейшим инструментом, позволяющим решать разноплановые задачи различных сложностей, как по обработке коллекций, так и по работе с единичными объектами, значительно сокращая количество программного кода и трудозатрат, необходимых для решения.

Наряду с непосредственным ядром .NET Framework, для реализации рассматриваемой информационной системы, будут использоваться следующие прикладные технологии:

1) ASP.NET Web API 2 – компонент .NET, предоставляющий возможность построения веб-сервисов, использующих архитектурный стиль REST (Representation State Transfer, «передача состояния представления»). Данная технология имеет удобную систему маршрутизации, фильтрации и обработки запросов, а позволяет организовать достаточно высокий уровень безопасности системы. Web API-сервисы могут совместно использоваться приложениями абсолютно различных типов – desktop, web, мобильными приложениями;

2) Microsoft Entity Framework – ORM-система (Object Relational Mapping, объектно-реляционное представление), предоставляющая разработчикам возможность работы с данными таблиц баз данных, спроецированных на объекты доменной модели. Это обеспечивает гораздо более высокий уровень абстракции, простоту реализации функциональности, обеспечивающей взаимодействие с хранилищами данных, значительно сокращая, при этом, объемы кодовой базы, необходимой для работы с ними по сравнению с традиционными подходами к разработке в данной области;

3) Microsoft ASP.NET Identity – платформа, позволяющая реализовывать функциональность аутентификации и авторизации пользователей, обеспечивающая достаточно гибкую работу с пользователями веб-приложений;

4) Microsoft Internet Information Services (IIS) – набор сервисов для различных служб интернета, поставляемый компанией Microsoft. В описываемом проектном решении, данный программный продукт будет использоваться для организации хостинга реализованного продукта;

5) ClosedXML – бесплатная open-source-библиотека, призванная предоставить .NET-разработчикам возможность простого, удобного и одновременно с этим гибкого создания и обработки документов-электронных таблиц Microsoft Office Excel в объектно-ориентированной манере, без непосредственной манипуляции XML-содержимым данных файлов. В рамках рассматриваемой информационной системы, данная библиотека будет использоваться для генерации отчетной документации.

Определив основной технологический стек для реализации серверной части системы, необходимо представить набор технологий, использование которых предполагается для разработки её клиентской части.

3.2.2 Определение набора технологий для реализации клиентского программного обеспечения АИС

Для реализации клиентской части системы, было принято решение об использовании движка графического представления Razor, являющегося частью платформы ASP.NET MVC. Использование Razor даёт преимущественную выгоду, поскольку, являясь частью .NET-платформы позволяет с ней интегрироваться, использовать большое количество вспомогательных классов, методов и механизмов, предоставляемых .NET для создания web-интерфейсов, значительно упрощая их разработку и сокращая количество затрачиваемого времени.

В дополнение к ASP.NET Razor, для манипуляций контентом страницы на стороне клиента, придания интерфейсу системы презентабельного внешнего вида, а также выполнения обращений к сервису бизнес-логики, получения

результатов и их обработки будет использоваться несколько дополнительных библиотек и технологий:

1) KnockoutJS – JavaScript библиотека, которая позволяет создавать сложные пользовательские интерфейсы, оставляя код расширяемым, читабельным и синтаксически чистым. В числе ее преимуществ – отслеживание зависимостей, декларативное связывание, кроссбраузерность, совместимость, компактность, а также возможность написания легкорасширяемого кода. Наконец, данная библиотека имеет достаточно хорошую документацию и не зависит от каких-либо других библиотек [20];

2) JQuery – библиотека, фокусирующаяся на взаимодействии JavaScript и HTML. Она представляет собой удобную и легковесную обертку для работы с DOM API. Помимо этого, она содержит в себе множество различных компонентов для повторного использования;

3) JQuery AJAX – Asynchronous Javascript and XML – технология, позволяющая клиентским приложениям веб-среды взаимодействовать с различного рода сервисами. Основным преимуществом данной технологии является её асинхронность, что означает отсутствие необходимости полной перезагрузки страниц, а также блокировки исполняющего потока на стороне клиента до получения результатов запроса;

4) Twitter Bootstrap – свободный набор инструментов для создания веб-приложений. Он включает в себя наиболее часто используемые HTML- и CSS-шаблоны для разметки веб-форм, кнопок, блоков навигации, и так далее.

На данном этапе, набор технологий, используемых для непосредственной реализации клиентской и серверной части автоматизированной информационной системы определен, теперь, можно переходить непосредственно к рассмотрению возможных вариантов СУБД для организации хранилища информационного обеспечения системы и выбору конкретной, которая будет использоваться в целях разработки.

3.2.3 Выбор СУБД и обоснование выбора

В настоящее время, системы управления базами данных являются неотъемлемым средством для организации, ведения, управления и поддержки баз данных автоматизированных информационных систем. Её использование значительно упрощает данные процессы, позволяет минимизировать количество непредвиденных ошибок различного уровня, которые могут возникнуть в результате выполнения операций над данными.

Одной из наиболее важных, преимущественных и актуальных причин, определяющих необходимость использования СУБД, является возможность осуществления контроля целостности данных, поддержки транзакций, а также большого количества встроенных функций и механизмов, позволяющих содержать данные в консистентном виде. Именно по этой причине, следует особое внимание уделить грамотному, осознанному и обоснованному подходу к её выбору, ведь правильный выбор СУБД является одним из залогов ее успешного функционирования, а также дальнейшего обслуживания и сопровождения.

Следует отметить, что выбранная СУБД должна обеспечивать полную совместимость с набором технологий, выбранным для реализации элементов автоматизированной информационной системы.

Для того, чтобы определить СУБД, наиболее удовлетворяющую потребностям разработки, были выбраны несколько наиболее популярных на сегодняшний день продуктов и произведен их сравнительный анализ по ряду критериев. Особое внимание в ходе сравнения придавалось удобству интеграции с технологиями, определенными в качестве используемых для разработки серверной части системы.

Результаты сравнительного анализа данных программных продуктов представлен в таблице 3.1.

Таблица 3.1 – Сравнительная характеристика СУБД

Наименование критерия	Microsoft SQL Server 2016	Oracle Database	PostgreSQL
1	2	3	4
Реализация базовой функциональности языка SQL	+	+	+
Наличие бесплатной версии	+	-	+
Контроль целостности базы данных	+	+	+
Наличие функционально насыщенного пользовательского интерфейса для управления и обслуживания базы данных	+	+	-
Поддержка механизма транзакций	+	+	+
Простота администрирования	+	+	-
Высокое качество документации	+	+	+
Поддержка кроссплатформенности	+	+	+
Совместимость с .NET Framework	+	+	+
Оптимизация под работу с .NET Framework	+	-	-
Простота настройки для работы с ORM Entity Framework	+	-	-
Интеграция с IDE Microsoft Visual Studio	+	-	-
Итого:	11	8	7

На основании приведенных выше результатов сравнительного анализа достоинств и недостатков каждого из выбранных продуктов, было принято решение об использовании Microsoft SQL Server 2016, поскольку она имеет ряд значительных преимуществ по сравнению с аналогами, а также полностью удовлетворяет всем нуждам разработки.

Выводы по параграфу 3.2

Определение технологического обеспечения проекта является одним из первых и наиболее важных шагов на пути к её непосредственной реализации. В параграфе 3.2 был представлен набор технологий, предполагаемых к использованию для удовлетворения потребностей разработки, описаны преимущества и выгоды, получаемые от использования данных технических средств. Следующим шагом к получению искомого программного продукта является непосредственная физическая реализация его компонентов.

3.3 Техническая реализация компонентов АИС

3.3.1 Физическая модель БД АИС

В п. 2.5 была представлена и описана логическая модель данных информационной системы, а в п. 3.2.3 произведен обоснованный выбор системы управления базами данных. Теперь необходимо построить физическую модель базы данных.

Физическая модель базы данных рассматривает основные отношения, организацию файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, отражение аспектов реализации объектов, выделенных в процессе построения логической модели данных в виде объектов конкретной базы данных: сущностей - в виде таблиц, их экземпляров – в виде строк данных таблиц, а атрибутов, соответственно в виде столбцов, а также все связанные с этим ограничения целостности и средства защиты.

Физическая модель данных разрабатываемой информационной представлена на рисунке 3.2.

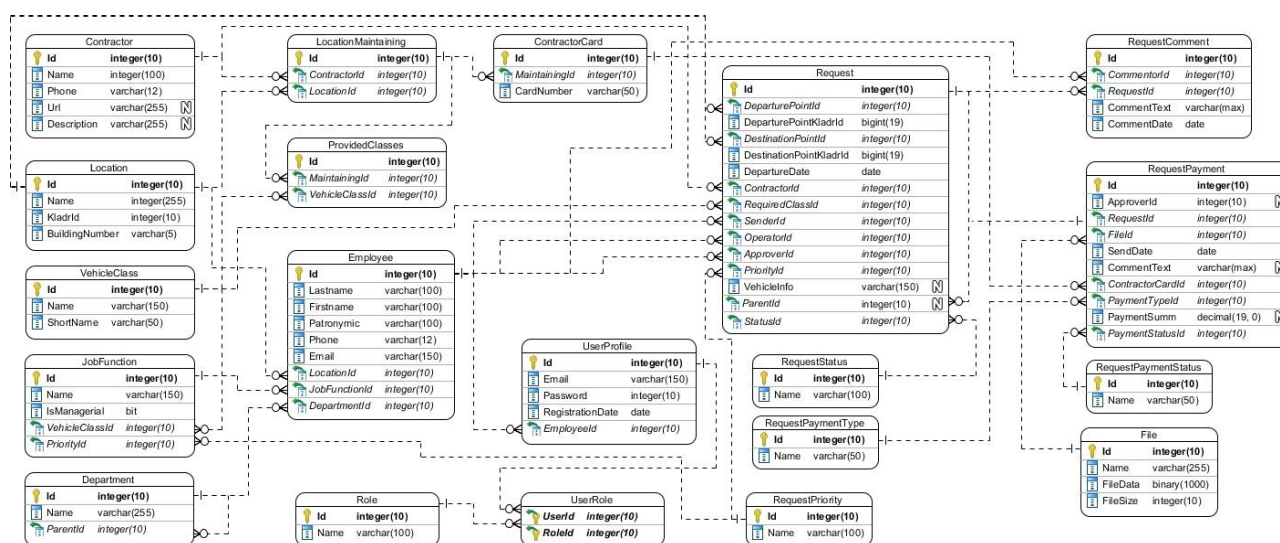


Рисунок 3.2 – Физическая модель базы данных

Как можно видеть на рисунке 3.2, построенная физическая модель отражает таблицы базы данных, с наборами атрибутов, каждый из которых имеет свой строго определенный тип с соответствующими ограничениями для тех полей, для которых это необходимо. Помимо этого, в каждой из таблиц

определены первичные и внешние ключи, а также установлены отношения между ними.

3.3.2 Описание программных модулей АИС

В сфере разработки на платформе .NET Framework, наиболее абстрактным понятием, схожим понятию «модуль» является понятие «сборка». Сборка представляет собой набор типов, интерфейсов и компонентов, сгруппированных по определенному обобщающему признаку, реализующих определенную функциональность, которые можно подключить к какому-либо другому проекту для дальнейшего использования реализуемой составляющими её классами и компонентами функциональности. Фактически, сборка представляет собой библиотеку классов (.dll). На этапе разработки, сборка представляет собой проект, являющийся частью решения, наполняемый различными программными единицами. Структуру решения рассматриваемой информационной системы можно представить набором сборок, представленным на рисунке 3.3.

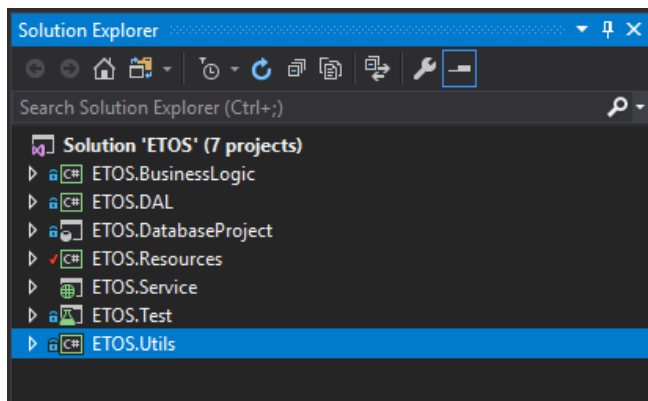


Рисунок 3.3 – Проектный состав решения

Как можно видеть, решение ETOS (сокр. от Employee Transport Ordering System) представляется следующим набором проектов:

1. ETOS.DatabaseProject – проект базы данных Microsoft SQL Server, содержит в себе скрипты создания таблиц базы данных, а также post-deployment скрипты, выполняемые непосредственно после развертывания базы данных на

сервере, осуществляющие заполнение таблиц-статических справочников необходимой информацией.

2. ETOS.DAL – Библиотека классов, предоставляющая интерфейсы, а также имплементирующую их функциональность, направленную на доступ к данным. Данная сборка содержит интерфейсы и реализации репозиториев, контекста хранилища данных, а также непосредственно саму модель данных, представленную в виде сущностей, проецируемых на базу данных при помощи механизма объектно-реляционного представления, реализуемого Microsoft Entity Framework.

3. ETOS.BusinessLogic – библиотека классов, содержащая интерфейсы, а также имплементирующие их типы, реализующие основную функциональность бизнес-логики, такую, например, как обработка заявок, отправка уведомлений и т.д., а также различные вспомогательные классы подобного плана.

4. ETOS.Service – непосредственно, сам проект сервиса бизнес-логики. Реализует в себе MVC-контроллеры, организующие маршрутизацию внутри веб-приложения, сверстанные шаблоны страниц, передаваемые клиенту для отображения, а также api-контроллеры, к которым асинхронно обращается клиентское приложение. Точка входа в серверную часть приложения. Содержит в себе конфигурацию MVC и web-api маршрутизации, бандлов, контейнера зависимостей для их дальнейшего разрешения.

5. ETOS.Resources – библиотека, содержащая в себе используемые ресурсы такие, например, как шаблоны высылаемых писем и т.д.

6. ETOS.Utills – вспомогательная библиотека, содержащая в себе имплементацию вспомогательных классов, повсеместно используемых всеми другими сборками.

7. ETOS.Test – проект, имплементирующий модульные тесты, проверяющие работоспособность реализованной в других сборках функциональности.

Теперь, когда проектный состав решения описан, можно перейти к непосредственному описанию обобщенного процесса сбора, передачи, обработки, а также выдачи информации.

3.3.3 Обобщенное описание процесса сбора, передачи, обработки и выдачи информации

В п. 3.1 была представлена и описана архитектурная модель рассматриваемой информационной системы, предполагающая отделение данных, логики, обрабатывающей их, а также уровня представления, предназначенного для отображения этих данных пользователю друг от друга. Суть предполагаемого подхода к процессу сбора, передачи и обработки, а также получения циркулирующей информации в контексте рассматриваемой системы представлена на рисунке 3.4. Как можно видеть на рисунке, система содержит в себе MVC-контроллеры, используемые для маршрутизации и отправки первоначального представления необходимой страницы пользователю системы. Каждая из страниц имеет в своем составе непосредственно HTML-верстку, используемые наборы CSS-стилей и JavaScript-библиотек, а также *модель представления*, содержащую в себе как все необходимые поля для хранения вводимых или отображаемых пользователю данных, так и набор простейших операций, выполняемых на стороне клиента: валидацию вводимых значений, отправку запросов на сервер приложений, а также получение и обработку их результатов.

Сервер приложений представлен в виде набора api-контроллеров (application programming interface), также имеющих собственную маршрутизацию, и инкапсулирующих в себе набор необходимых для их работы *слабосвязанных компонентов*. Под *слабой связанностью* в данном контексте подразумевается использование не конкретной реализации какого-либо из необходимых компонентов, а только лишь его абстрактного интерфейса. Этим обеспечивается принцип *внедрения зависимостей* – один из пяти главных принципов объектно-ориентированного программирования и проектирования

SOLID. Внедрение осуществляется через конструктор, посредством сконфигурированного контейнера зависимостей.

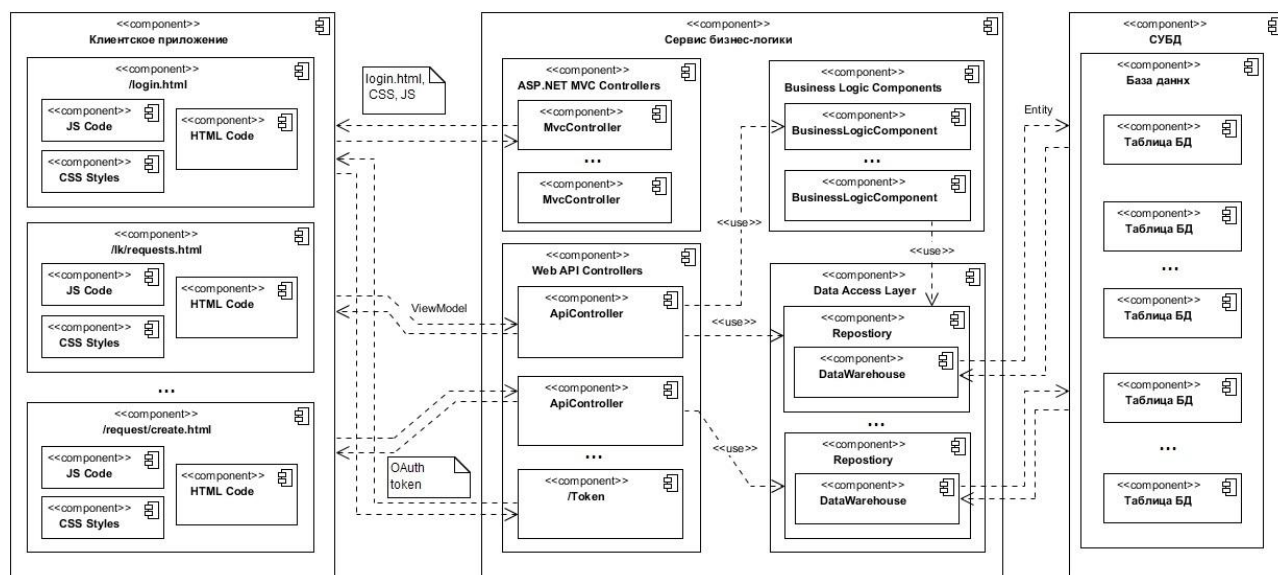


Рисунок 3.4 – Концептуальная схема процесса сбора, передачи, обработки и получения информации

Главное преимущество организации слабой связанности программных единиц и принципа внедрения зависимостей в том, что при таком подходе, каждый из компонентов и элементов приложения зависим лишь от абстракций, получаемых через конструктор, извне, а значит, каждая из зависимостей, обращение к которой осуществляется через её интерфейс становится легко заменяемой, обеспечивает гибкость программного кода, его тестируемость, легкую замену одного такого компонента другим. Следует отметить, что данный подход используется на всех уровнях системы.

Каждый из апи-контроллеров предоставляет собственный *контракт данных и операций*, который каждый из вызывающих его клиентов обязан соблюдать. Получив запрос на вход, сервер приложений, благодаря сконфигурированным механизмам маршрутизации WebAPI находит метод, тип и имя которого совпадает с вызываемым. После этого, механизм связывания позволяет осуществить передачу передаваемых в запросе данных апи-модели на ту сущность, которую ожидает выбранный конвейером WebAPI для исполнения метод. Непосредственно после этого, начинается выполнение самого метода.

В каждом из своих ответов, помимо необходимых данных, арі-контроллер также передает соответствующий HTTP-код, определяющий статус выполнения операции. Так, например, 403 код (Forbidden) возвращается при указании неверного логина или пароля пользователя и невозможности его авторизации, а также, при попытке получения той информации к которой пользователь не имеет соответствующего доступа, код 500 (Internal Server Error) может получаться клиентом при возникновении непредвиденных ошибок, или же код 200 (OK) отправляется сервером случае успешно выполненной операции.

При отсутствии каких-либо ошибок валидации данных, арі-контроллер начинает непосредственную обработку или получение данных. Промежуточным звеном в работе с данными является специальный компонент, реализующий паттерн *Репозиторий*, являющийся посредником между слоем доступа к данным и непосредственно доменным слоем, представленным в виде сущностей-объектов, каждая из которых, благодаря механизмам Microsoft Entity Framework отражается на соответствующую таблицу в хранилище данных, представляя собой одну её запись. Следует заметить, что, поскольку, доменная сущность и арі-модель, физически являются не одним и тем же, перед непосредственной манипуляцией сущностью БД (в случае добавления, обновления данных и т.д.), или же, наоборот, получения сущностей из БД для последующей отправки их на клиента, система использует механизм мэппинга, фактически сводящийся к переносу данных из объектов-сущностей в объекты-модели и наоборот. Использование данного подхода обосновано тем, что далеко не всегда и не каждому представлению необходимы все данные, имеющиеся в сущности. Кроме того, может возникнуть такая ситуация, когда модель представления, используемая клиентской страницей может использовать агрегированные данные, собранные из нескольких сущностей, организованные в необходимом и достаточном для представления объеме, и виде. Наконец, использование такого подхода обеспечивает защищенность и консистентность данных, хранящихся в БД. Фактически, арі-модель

предназначена лишь для передачи данных между клиентом и сервером, представляет собой реализацию паттерна *Data Transfer Object*.

Как уже говорилось ранее, система предоставляет компоненты, именуемые репозиториями, основная задача которых – выполнять простейшие операции с сущностями в базе данных (получение, обновление, удаление, выборки и т.д.). Каждый из таких объектов имеет в своем составе компонент, представляющим собой контекст используемого хранилища данных. Фактически, именно внутри этого компонента происходит непосредственное связывание приложения с хранилищем его данных. Благодаря тому, что репозитории используют обращаются к контексту хранилища через интерфейс, а не его фактическую реализацию, он также может быть заменен на любой другой тип хранилища: файловое хранилище или, даже, *in-memory* коллекцию, важно лишь чтобы используемый компонент имплементировал методы, объявляемые интерфейсом.

Подводя итог, хотелось бы еще раз подчеркнуть то, что такой подход позволяет обеспечивать гибкость архитектуры, обеспечиваемую слабой зависимостью составляющих её компонентов друг от друга, лёгкой заменой одного компонента на другой, тестируемостью кодовой базы, отсутствием смешения логики.

Рассмотрим суть данного подхода к организации взаимодействия клиента и сервера на примере реализации механизма аутентификации и авторизации пользователей внутри системы.

Страница авторизации определяет форму для ввода логина и пароля, фрагмент которой представлен представленную в листинге 3.1. Также, данная страница реализует модель представления, которая, благодаря механизмам связывания KnockoutJS хранит в себе данные, а также отслеживает их изменения и обновляет свое содержимое, а также функцию, реализующую отправку данных на сервер приложений, а затем обработку получаемых результатов.

В данном случае, форма авторизации вызывает специальный сервис, предоставляемый компонентами ASP.NET Identity, главное назначение которого – формирование и выдача пользователю токена авторизации в том случае, если имя пользователя и пароль были введены корректно, или же, отправка сообщения об ошибке клиенту в том случае, если введенные логин-пароль являются некорректными. Модель представления формы авторизации представлена в листинге 3.2.

```
<form data-bind="submit: submitHandler, with: credentials">
  <div style="margin-bottom: 25px" class="form-group">
    <label>Ваш E-mail:</label>
    <div class="input-group">
      <span class="input-group-addon">
        <i class="glyphicon glyphicon-user"></i>
      </span>
      <input type="text" name="login" data-bind="value: userName" class="form-
        control" placeholder="Ваш логин"/>
    </div>
  </div>
  <div style="margin-bottom: 15px" class="form-group">
    <label>Ваш пароль:</label>
    <div class="input-group">
      <span class="input-group-addon">
        <i class="glyphicon glyphicon-lock"></i>
      </span>
      <input type="password" name="password" data-bind="value: password" class="form-
        control" placeholder="Ваш пароль"/>
    </div>
  </div>
</form>
```

Листинг 3.1 – Фрагмент HTML-верстки страницы авторизации

Как можно видеть на листингах 3.1 и 3.2, в качестве обработчика события нажатия кнопки, имеющей тип “submit”, расположенной на форме авторизации указывается функция submitHandler(), реализуемая моделью представления. Соответственно, при нажатии на данную кнопку, клиентское приложение сериализует содержимое объекта credentials в JSON и отправляет AJAX-запрос к серверу приложений, а конкретнее, к его специальному сервису, расположенному по адресу /Token, листинг исходного кода которого представлен в приложении Б.

После получения запроса сервисом бизнес-логики, осуществляется валидация модели на стороне сервера. Несмотря на наличие валидации на стороне клиента, наличие серверной валидации так же, является необходимым,

обосновано это обособленным функционированием сервера приложения, отсутствием возможности определенно гарантировать валидность данных проходящих api- методам на вход, поскольку отправителем запроса может быть любое аутентифицированное устройство и приложение, непосредственно имеющее к нему сетевой доступ.

```
function AuthorizationModel() {
    var self = this;
    self.credentials = {
        grant_type: 'password',
        userName: ko.observable(null)
        .extend({
            required: true,
            message: 'Заполните'
        }),
        password: ko.observable(null)
        .extend({
            required: true,
            message: 'Заполните'
        }),
        isPersistent: ko.observable(false)
    }
    self.submitHandler = function() {
        const userInfo = ko.toJS(credentials);
        var tokenKey = "tokenInfo";

        $.ajax({
            type: 'POST',
            url: '/Token',
            data: userInfo,
            success: function(response) {
                sessionStorage.setItem(tokenKey, response.access_token);
                window.location = "lk/requests";
            },
            error: function(jqXHR) {
                var msg = '';

                if (jqXHR.responseJSON.error === "invalid_grant") {
                    msg = 'Неправильный логин или пароль.';
                } else {
                    msg = 'Произошла непредвиденная ошибка.';
                }

                $("#page-title").notify(msg, { position: "bottom center" });
            }
        });
    };
};

var authorizationModel = new AuthorizationModel();
ko.applyBindings(authorizationModel);
```

Листинг 3.2 – Модель представления формы авторизации

В том случае, если модель считается валидной, сервис вызывает специальный объект, реализующий поиск зарегистрированного с указанным

логином и паролем пользователя и, при его наличии, получение токена авторизации для него. Примерный вид токена, получаемого в виде успешной аутентификации, представлен в листинге 3.3

```
{
  access_token: "u3X0CYV91f2P6odbceNIY_BnkfSpN7gQwzksRi_...0iRP1HYNMEES9",
  token_type: "bearer",
  expires_in: 1209599,
  userName: "employee@etos.com",
  .issued: "Sat, 07 Mar 2018 21:42:16 GMT",
  .expires: "Sat, 21 Mar 2018 21:42:16 GMT"
}
```

Листинг 3.3 – Вид получаемого токена аутентификации

Преимущество использования механизмов аутентификации на основе токенов заключается в организации механизмов предоставления доступа для большого количества устройств, даже не имеющих возможность работы с cookie, которые, как правило, используются для традиционной web-forms-based аутентификации.

Выводы по параграфу 3.3

В параграфе 3.3 была представлена физическая модель хранилища данных, а также технический состав реализованных компонентов, описан обобщенный технологический рабочий поток системы, обозначены зависимости и точки соприкосновения её модулей. Поскольку, данный параграф завершает описание непосредственного процесса проектирования и разработки элементов автоматизированной информационной системы, далее, необходимо представить описание основных аспектов работы реализованной функциональности.

3.4 Описание реализованной функциональности

Рассмотрим реализованную функциональность на примере авторизации, создания заявки и обработки её офис-менеджером.

Работа с системой всегда начинается с обращения к какой-либо из страниц клиентского приложения. И, поскольку, доступ к системе регулируется

провайдерами аутентификации, неавторизованному пользователю нужно будет осуществить вход в систему, на странице, содержащей форму авторизации, вид которой представлен на рисунке 3.5.

Данная форма располагается по ссылке `http://адрес_хоста:[порт]/login`. Неавторизованные пользователи также перенаправляются на данную страницу при попытке получения доступа к той части системы, доступ к которой для неавторизованных пользователей закрыт.

Войти в систему

Ваш E-mail:

employee@etos.com

Ваш пароль:

.....

☐ Запомнить меня

Войти Забыл пароль

Рисунок 3.5 – Вид формы входа в систему

Следует обратить внимание на то, что поля для ввода email и пароля валидируются. И в случае отсутствия ввода данных в одно из полей, оно будет подсвечено красным цветом, а также сопровождается дополнительным текстовым сообщением. Кроме того, кнопка входа в систему станет неактивной до тех пор, пока валидные с точки зрения клиентского приложения данные не будут введены. Данная ситуация показана на рисунке 3.6.

Войти в систему

Ваш E-mail: ↓ Заполните

@

Ваш пароль:

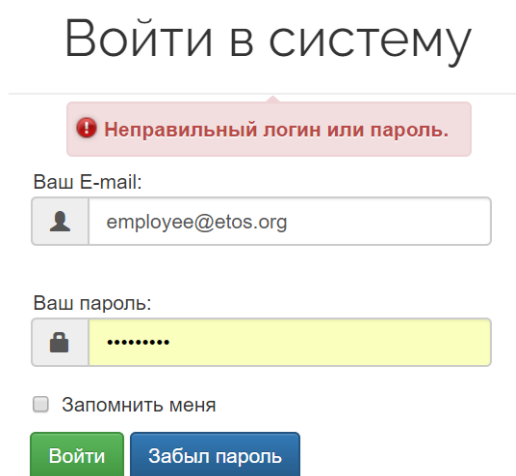
.....

☐ Запомнить меня

Войти Забыл пароль

Рисунок 3.6 – Вид формы авторизации при наличии валидационных ошибок

При отправке данных авторизации сервису бизнес-логики, осуществляется проверка на наличие существующего пользователя с указанным паролем, и при его отсутствии, сервис бизнес-логики (далее – СБЛ) отправляет клиентскому приложению сообщение об ошибке. Данная ситуация показана на рисунке 3.7.



The screenshot shows a login form titled "Войти в систему" (Log in to the system). At the top, there is a red error message box that says "Неправильный логин или пароль." (Incorrect login or password.). Below this, there are two input fields: "Ваш E-mail:" (Your email) with the value "employee@etos.org" and "Ваш пароль:" (Your password) with masked characters ".....". There is a checkbox labeled "Запомнить меня" (Remember me) which is unchecked. At the bottom, there are two buttons: "Войти" (Log in) in green and "Забыл пароль" (Forgot password) in blue.

Рисунок 3.7 – Вид формы при некорректно введенном логине или пароле

Поскольку, код клиентского приложения умеет реагировать на ошибки, получаемые от СБЛ, он выводит соответствующую всплывающую нотификацию с информацией об ошибке.

В том случае, если введенные пользователем авторизационные данные корректны, он перенаправляется на страницу отображения личных заявок. В настоящий момент, страница «Мои заявки» содержит две вкладки – «Прошедшие», которая отображает завершенные и закрытые заявки, а также вкладку «Текущие», вид которой представлен на рисунке 3.8.

Как можно видеть на рисунке 3.8, в настоящий момент, навигационное меню содержит ссылки на страницу «Кабинет», «Мои заявки», «Выйти», а также кнопку «Создать заявку».

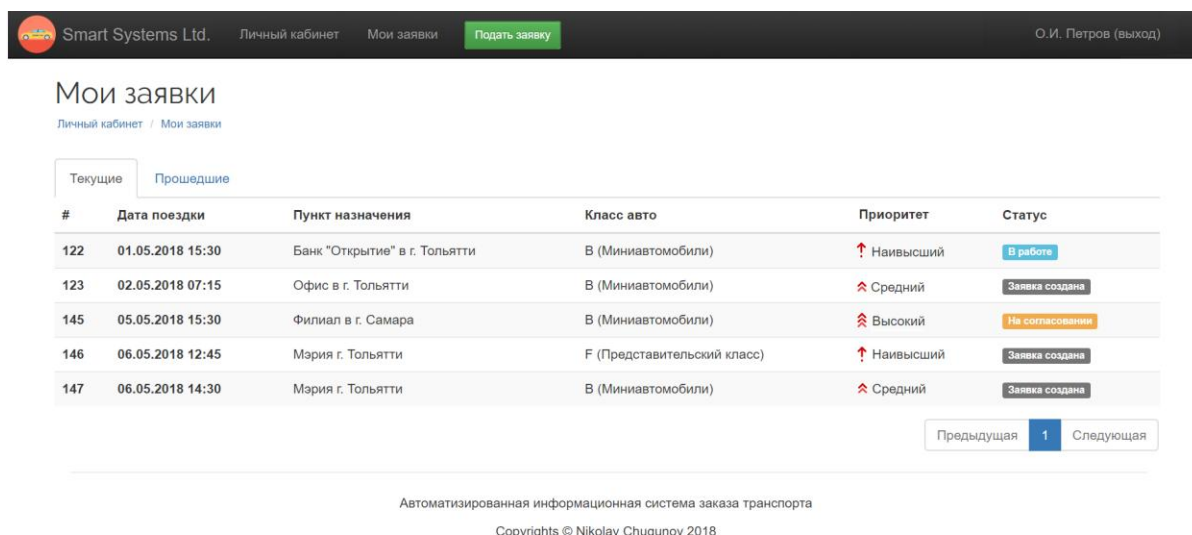


Рисунок 3.8 – Вид вкладки «Текущие» страницы «Мои заявки»

Вид вкладки «Прошедшие» представлен на рисунке 3.9.

Мои заявки

Личный кабинет / Мои заявки

Текущие Прошедшие

#	Дата поездки	Пункт назначения	Класс авто	Приоритет	Статус
35	06.05.2018 14:30	Мэрия г. Тольятти	В (Миниавтомобили)	↗ Средний	Закрыта
33	05.05.2018 07:30	Офис	В (Миниавтомобили)	↑ Наивысший	Закрыта
32	03.05.2018 18:00	70 Лет Октября ул., д. 12	А (Микроавтомобили)	↗ Средний	Закрыта
28	22.04.2018 10:50	Филиал в г. Самара	С (Европейский "средний класс")	↗ Высокий	Закрыта
22	18.04.2018 23:00	70 Лет Октября, ул., д. 12	А (Микроавтомобили)	↑ Наивысший	Закрыта
20	10.04.2018 07:30	Отдел ФНС в г. Тольятти	В (Миниавтомобили)	↗ Средний	Закрыта
17	07.04.2018 18:00	70 Лет Октября ул., д. 12	В (Миниавтомобили)	↑ Наивысший	Завершена неуспешно
12	01.04.2018 07:15	Офис в г. Тольятти	А (Микроавтомобили)	↗ Средний	Закрыта
5	27.03.2018 15:30	Филиал в г. Самара	В (Миниавтомобили)	↗ Высокий	Закрыта
3	27.03.2018 11:30	Мэрия г. Тольятти	F (Представительский класс)	↑ Наивысший	Закрыта

Предыдущая 1 2 3 Следующая

Рисунок 3.9 – Вид вкладки «Прошедшие» страницы

При нажатии на кнопку «Создать заявку» соответственно, открывается форма создания заявки, вид которой представлен на рисунке 3.10.

При отправке и сохранении новой заявки, офис-менеджеру приходит соответствующее e-mail оповещение, представленное на рисунке 3.11.

В личном кабинете, офис-менеджеру, помимо вкладок, представленных добавленная заявка будет отображаться в списке заявок, ожидающих

обработки. При открытии, офис-менеджеру отображается страница с заявкой, вид которой представлен на рисунке 3.12.

Smart Systems Ltd. Личный кабинет Мои заявки **Подать заявку** О.И. Петров (выход)

Создать заявку

ФИО сотрудника: Петров Олег Сергеевич

Приоритет выполнения: Высокий

Пункт отправки: Офис ООО "Смарт Системз" в г. Москва

Адрес пункта отправки: г. Москва, ул. Вышинская, д. 54

Пункт назначения:

Адрес пункта назначения: г. Москва, ул. Кремлевская, д.5

Предполагаемая дата отправки: 04.05.2018 15:45

Необходимый класс автомобиля: Миниавтомабили

Отправить заявку

Автоматизированная информационная система заказа транспорта
Copyrights © Nikolay Chugunov 2018

Рисунок 3.10 – Вид страницы «Создать заявку»

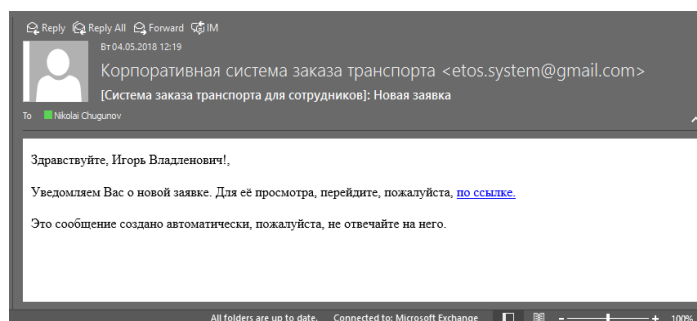


Рисунок 3.11 – Вид email-оповещения о новой заявке

Заявка № 155 от 04.05.2018 **Взять в работу** **Отправить на согласование** **Закрыть** **Заявка создана**

ФИО сотрудника: Петров Олег Сергеевич
Номер телефона: +7 (935) 213-42-58
Email: oleg_petrov@smartsys.com

Необходимый класс транспорта: А (Миниавтомабили)
Приоритет: Высокий
Предполагаемые дата и время поездки: 04.05.2018 15:45

Пункт отправки: Офис ООО "Смарт Системз" в г. Москва
Адрес пункта отправки: г. Москва, ул. Вышинская, д.54

Пункт назначения: -
Адрес пункта назначения: г. Москва, ул. Кремлевская, д.5

Рисунок 3.12 – Вид страницы отображения заявки

При нажатии кнопки «Взять в работу», статус заявки переходит в статус «В работе», а пользователю, оформившему заявку, высылается соответствующее уведомление, вид которого представлен на рисунке 3.13.

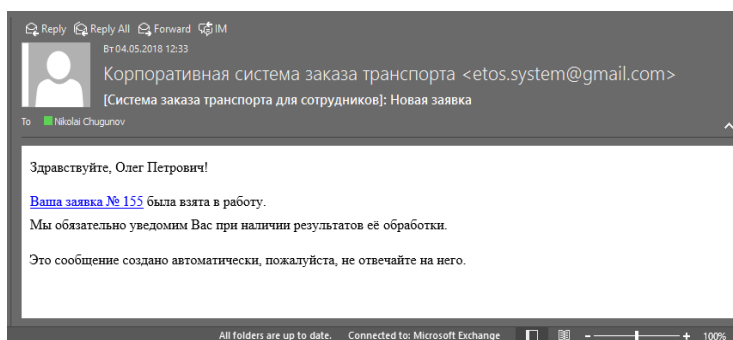


Рисунок 3.13 – Вид уведомления о передаче заявки в работу

При начале работы с заявкой, офис-менеджеру становится доступным компонент формы выбора подрядчика и указания деталей. Вид формы представлен на рисунке 3.14.

The image shows a web form titled 'Заявка № 155 от 04.05.2018'. At the top right are buttons 'Отправить на согласование' and 'Закрыть', and a 'В работе' status indicator. The form is divided into two columns. The left column contains: 'ФИО сотрудника: Петров Олег Сергеевич', 'Номер телефона: +7 (935) 213-42-58', 'Email: oleg_petrov@smartsys.com', 'Пункт отправки: Офис ООО "Смарт Системз" в г. Москва', and 'Адрес пункта отправки: г. Москва, ул. Выхинская, д.54'. The right column contains: 'Необходимый класс транспорта: А (Миниавтомобили)', 'Приоритет: Высокий', 'Предполагаемые дата и время поездки: 04.05.2018 15:45', 'Пункт назначения: -', and 'Адрес пункта назначения: г. Москва, ул. Кремлевская, д.5'. Below this is a section 'Детали поездки' with two sub-sections. 'Подрядчик' has a dropdown menu showing 'Такси Амбассадор' and a 'Выбрать' button. 'Детали транспортного средства' has a text input field. Under 'Вид оплаты', there are two buttons: 'Корпоративная карта' (highlighted in green) and 'Оплата из личных средств'. Below these is a '№ Карты' dropdown menu showing '№458'. At the bottom left is a green 'Сохранить' button.

Рисунок 3.14 – Вид формы обработки заявки

При нажатии кнопку «Выбрать подрядчика», отображается модальное окно выбора, вид которого представлен на рисунке 3.15.

После выбора подрядчика и указания деталей предстоящей поездки, сотруднику, подавшему заявку, высылается соответствующее уведомление, вид которого представлено на рисунке 3.16.



Рисунок 3.15 – Вид модального окна выбора подрядчика

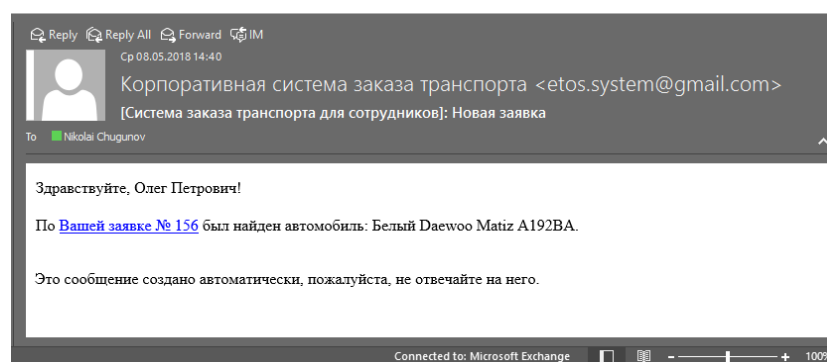


Рисунок 3.16 – Вид уведомления о нахождении транспорта для выполнения заявки

В заключении, хотелось бы сказать, что, несмотря на то, что представленные разработанные компоненты описывают не полный процесс, и, возможно, имеют дизайн, выраженный не идеальным образом, тем не менее, они позволяют интуитивно ориентироваться по графическому интерфейсу и понять основные способы взаимодействия пользователя с системой.

Выводы по главе 3

В третьей главе, было описано технологическое обеспечение, используемое для реализации элементов информационной системы заказа транспорта для сотрудников компаний, определено архитектурное решение, взятое за основу реализации системы, представлено описание специфики взаимодействия её компонентов между собой, приведены примеры данного взаимодействия, отражен чек-лист тестовых сценариев, направленных на, осуществление проверки корректной работоспособности основной

функциональности реализованных компонентов. Наконец, было представлено описание реализованной функциональности. Поскольку, процесс разработки на данном этапе завершен, необходимо подвести его итоги, оценить достоинства и недостатки реализованной функциональности, определить дальнейшие пути развития проекта.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе, были рассмотрены вопросы, описывающие процесс разработки компонентов информационной системы заказа транспорта для сотрудников компаний, реализующая в себе функциональность, позволяющую использующим её пользователям:

- 1) отправлять заявки на заказ корпоративного транспорта, при наличии такой потребности;
- 2) рассматривать и обрабатывать заявки ответственным за данные процессы лицам;
- 3) осуществлять выборки по определенным условиям за указанный период и генерирующая результаты выборки в виде документа электронных таблиц Microsoft Office Excel;
- 4) отправлять скан-копии документов, подтверждающие оплату заявок сотрудниками из собственных средств на рассмотрение;
- 5) получать от системы уведомления в виде email-писем.

Заложенная в основу архитектурная модель системы является достаточно гибкой, позволяет расширять и дополнять имеющуюся кодовую базу с минимальными рисками. Это было достигнуто путем организации слабой связанности – зависимостей компонентов только от абстракций. Клиентское приложение максимально отвязано от бизнес-операций, и, поскольку, сервис бизнес-логики представлен в виде независимых api-контроллеров, при соблюдении контрактов данных и операций, это дает преимущественную возможность использования сервиса сторонними, например, мобильными приложениями.

Однако, поскольку, разработка была направлена только на реализацию компонентов системы, её начального приближения, следует также обозначить ряд планов дальнейшего развития и доработки системы:

- 1) разработка системы логгирования – ведения журнала операций, действий и ошибок, возникающих в ходе жизненного цикла работы;

2) реализация функциональности, запланированной на версию 1.1 и 1.2, перечисленной в требованиях к разработке;

3) доработка и рестайлинг клиентского приложения информационной системы, рефакторинг его исходного кода;

4) реализация начального прототипа мобильного приложения, использующего уже существующий сервис бизнес-логики, а также реализуемую им функциональность.

Подводя итог, можно сказать, что несмотря на очевидную необходимость в продолжении разработки, наращивании функциональной и документальной базы, реализованные компоненты системы реализуют задачи, поставленные перед ними и представляют собой отличную технологическую основу для дальнейшего развития.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Учебники и учебные пособия

1. Албахари Дж., Албахари Б. C# 5.0. Справочник. Полное описание языка.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2014. – 1008 с.
2. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. – 496 с.: ил. ISBN 5-94074-334-X.
3. Вигерс, К. Разработка требований к программному обеспечению. 3-е изд., дополнительное / К. Вигерс, Д. Битти., Пер. с англ. – М.: Издательство «Русская редакция». - СПб.: БХВ-Петербург, 2014. – 736 с.
4. Култыгин О. П. Администрирование баз данных. СУБД MS SQL Server: учебное пособие / О. П. Култыгин. - Москва : МФПА, 2012. - 232 с. : ил. - (Университетская серия). – ЭБС «IPRBooks».
5. Олейник П.П. Корпоративные информационные системы: Учебник для вузов. Стандарт третьего поколения / П.П. Олейник. – СПб.: Питер, 2012. – 176 с.: ил.
6. Маклафлин, Б. Объектно-ориентированный анализ и проектирование / Б. Маклафлин, Г. Поллайс, Д. Уэст. — СПб.: Питер, 2013. — 608 с.
7. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework на языке C#. 4-е изд. – СПб.: Питер, 2013. – 896 с.
8. Фаулер, Мартин. Архитектура корпоративных программных приложений / М. Фаулер.: Пер. с англ. – М.: Издательский дом «Вильямс», 2007. – 544 с.
9. Фрейн, Б. HTML5 и CSS3 = ResponsiveWebDesignwith HTML5 and CSS3: Разработка сайтов для любых браузеров и устройств / Б. Фрейн : [пер. с англ. В. Черник]. – Санкт-Петербург: Питер, 2014. – 298 с.
10. Фримен А., Раттц-мл., Джозеф С. LINQ: язык интегрированных запросов в C# 2010 для профессионалов.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 656 с.

11. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: учебник для вузов / под ред. А.Д.Хомоненко.- 3-е изд. - СПб: Корона-Принт, 2012.- 672 с.

12. Черемных С.В. Моделирование и анализ систем. IDEF-технологии: практикум / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2006. – 192 с.: ил. ISBN 5-279-02564-X.

Интернет-источники

13. Введение в Knockout – новый взгляд на построение Javascript – приложений [Электронный ресурс]: статья / Сайт Freaksidea. Режим доступа: <http://freaksidea.com/javascript/show-65-vvedenie-v-knockout---novyj-vzglyad-na-postroenie-javascript-prilozhenij>, свободный (дата обращения 31.03.2018).

14. Документ и объекты страницы [Электронный ресурс]: серия статей / Сайт Современный учебник JavaScript. Режим доступа: <http://learn.javascript.ru/document>, свободный (даты обращения 10.04.2018, 13.04.2018, 14.04.2018, 19.04.2018, 02.05.2018, 12.05.2018).

15. Доменный объект [Электронный ресурс]: статья / Сайт Википедия – свободная энциклопедия. Режим доступа: https://ru.wikipedia.org/wiki/Доменный_объект (дата обращения 02.05.2018)

16. Иерархическая организация [Электронный ресурс]: статья / Сайт Википедия – свободная энциклопедия. Режим доступа: https://ru.wikipedia.org/wiki/Иерархическая_организация, свободный (дата обращения 15.04.2018)

17. Нормативно-справочная информация [Электронный ресурс]: статья / Сайт Словари и энциклопедии на Академике. Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/302417>, свободный (дата обращения 01.05.2018)

18. Официальный сайт Bootstrap на русском языке [Электронный ресурс]: серия статей / Режим доступа: <http://mybootstrap.ru>, свободный (даты обращения 10.04.2018, 12.04.2018, 22.04.2018, 12.05.2018).

19. Правила пользования корпоративным такси [Электронный ресурс]: статья / Сайт Wheely – блог о корпоративных перевозках. Режим доступа: <https://wheely.com/blog/b2b/pravila-polzovaniya-korporativnym-taksi/>, свободный (дата обращения 02.04.2018).

20. Работа с данными и Ajax [Электронный ресурс]: серия статей / Сайт Smarly.net библиотека. Режим доступа: <http://smarly.net/pro-jquery/pro-jquery-data-and-ajax>, свободный (даты обращения 12.03.2018, 15.03.2018, 21.04.2018, 22.04.2018, 12.05.2018).

21. Руководство по ASP.NET Web API 2 [Электронный ресурс]: серия статей / Сайт о программировании METANIT.COM. Режим доступа: http://metanit.com/sharp/aspnet_webapi/, свободный (даты обращения: 06.02.2018, 05.04.2018, 13.04.2018, 20.04.2018, 07.05.2018).

22. Типовые организационные структуры предприятий [Электронный ресурс]: статья / Сайт Корпоративный менеджмент. Режим доступа: http://www.cfin.ru/management/iso9000/iso9000_orgchart.shtml, свободный (дата обращения 20.02.2018).

23. ASP.NET Identity и системы аутентификации [Электронный ресурс]: серия статей / Сайт о программировании на платформе .NET Framework Professorweb. Режим доступа: http://professorweb.ru/my/ASP_NET/identity/, свободный (даты обращения: 07.02.2018, 09.02.2018, 13.03.2018, 18.05.2018).

24. Jumpstart KnockoutJS with ASP.NET MVC/WebAPI [Электронный ресурс]: статья / Сайт Codeproject.com – For those who code. Режим доступа: <https://www.codeproject.com/Articles/755386/Jumpstart-KnockoutJS-with-ASP-NET-MVC-Web-API>, свободный (дата обращения: 19.05.2018).

Литература на иностранном языке

25. Bailis, P. Readings in Database Systems – 5th edition, MIT Press, 2015.

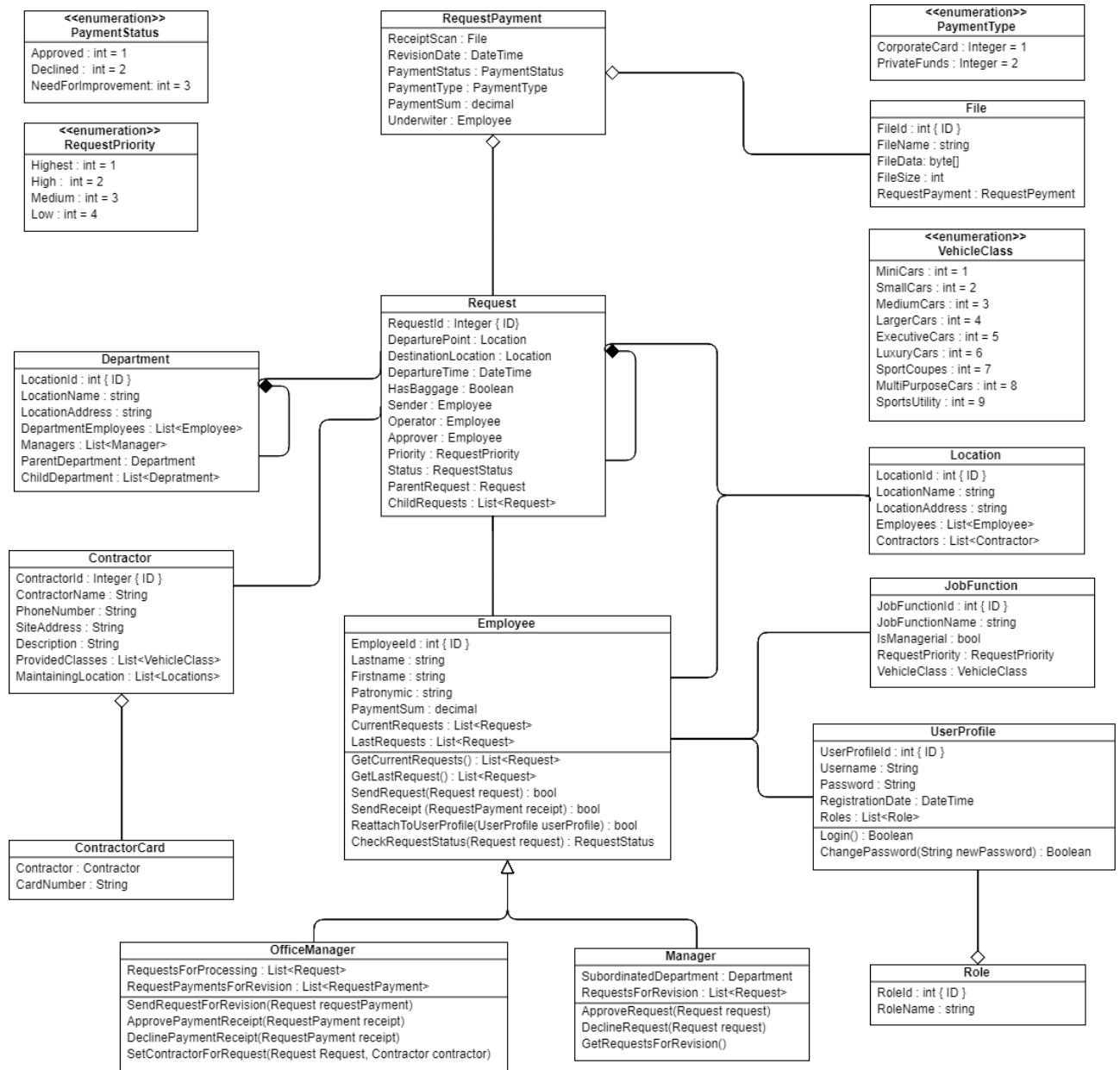
26. Esposito, D. Architecting Applications for the Enterprise – 2nd edition, Microsoft Press, 2014.

27. Haridi, S. Concepts, Techniques, and Models of Computer Programming – 3rd edition, MIT Press, 2014.

28. Kalloniatis C. Innovative Information Systems Modelling Techniques – InTech Press, 2016.

ПРИЛОЖЕНИЕ А

Доменная модель предметной области



ПРИЛОЖЕНИЕ Б

Листинг кода сервиса аутентификации

Модель представления для авторизации

```
using System.ComponentModel.DataAnnotations;

namespace ETOS.Service.Models.Account
{
    /// <summary>
    /// Описывает модель авторизации пользователя в системе.
    /// </summary>
    public class AuthorizationModel
    {
        /// <summary>
        /// Идентификатор пользователя.
        /// </summary>
        public string Id { get; set; }

        /// <summary>
        /// Email-адрес пользователя.
        /// </summary>
        [Required(AllowEmptyStrings = false, ErrorMessage = "Необходимо указать логин.")]
        public string Username { get; set; }

        /// <summary>
        /// Пароль пользователя.
        /// </summary>
        [Required(AllowEmptyStrings = false, ErrorMessage = "Необходимо указать пароль.")]
        public string UserPassword { get; set; }

        /// <summary>
        /// Флаг "Запомнить меня".
        /// </summary>
        public bool IsPersistent { get; set; }
    }
}

using Microsoft.AspNet.Identity;

using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OAuth;

using Owin;

using ETOS.DAL.DataAccess.Implementation;
using ETOS.DAL.DataAccess.Implementation.Identity;
using ETOS.Service.Providers;

namespace ETOS.Service
{
    public partial class Startup
    {
        /// <summary>
        /// Конфигурация OAuth-авторизации и аутентификации.
        /// </summary>
        public static OAuthAuthorizationServerOptions OAuthOptions { get; private set; }

        public static string PublicClientId { get; private set; }

        /// <summary>
```

```

    /// Реализует конфигурирование механизмов аутентификации и авторизации.
    /// </summary>
    /// <param name="app">Конфигуратор приложения.</param>
    public void ConfigureAuth(IAppBuilder app)
    {
        // Приложение должно использовать один экземпляр контекста БД и менеджера УЗ за
        // жизненный цикл приложения.
        app.CreatePerOwinContext(EtosDataWarehouseContext.Create);
        app.CreatePerOwinContext<UserProfileManager>(UserProfileManager.Create);

        app.UseCookieAuthentication(new CookieAuthenticationOptions { LoginPath = new
        PathString("/Login")});
        app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

        // Настройка OAuth-авторизации.
        PublicClientId = "self";
        OAuthOptions = new OAuthAuthorizationServerOptions
        {
            // Установка пути, по которому будет получаться токен.
            TokenEndpointPath = new PathString("/Token"),

            // Указывает на вышеопределенный провайдера.
            Provider = new ApplicationOAuthProvider(PublicClientId),

            // Срок жизни токена авторизации (в днях).
            AccessTokenExpireTimeSpan = TimeSpan.FromDays(14),
        };

        // Включает возможность использования токенов.
        app.UseOAuthBearerTokens(OAuthOptions);
    }
}

using System;
using System.Collections.Generic;
using System.Threading.Tasks;

using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OAuth;

using ETOS.DAL.DataAccess.Implementation.Identity;

namespace ETOS.Service.Providers
{
    /// <summary>
    /// Провайдер OAuth-аутентификации.
    /// </summary>
    public class ApplicationOAuthProvider : OAuthAuthorizationServerProvider
    {
        private readonly string _publicClientId;

        public ApplicationOAuthProvider(string publicClientId)
        {
            _publicClientId = publicClientId ?? throw new
            ArgumentException(nameof(publicClientId));
        }

        /// <summary>
        /// Реализует формирование токена или проброс ошибки вызывающей стороне, в случае
        /// указания некорректного логина и пароля.
        /// </summary>

```

```

        public override async Task
GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)
        {
            var userManager = context.OwinContext.GetUserManager<UserProfileManager>();

            var user = await userManager.FindAsync(context.UserName, context.Password);

            if (user == null)
            {
                context.SetError("invalid_grant", "Неверно указаны логин или пароль.");

                return;
            }

            var oAuthIdentity = await
user.GenerateUserIdentityAsync(userManager, OAuthDefaults.AuthenticationType);
            var cookiesIdentity = await user.GenerateUserIdentityAsync(userManager,
CookieAuthenticationDefaults.AuthenticationType);
            var properties = CreateProperties(user.UserName);
            var ticket = new AuthenticationTicket(oAuthIdentity, properties);

            context.Validated(ticket);

            context.Request.Context.Authentication.SignIn(cookiesIdentity);
        }

        public override Task TokenEndpoint(OAuthTokenEndpointContext context)
        {
            foreach (var property in context.Properties.Dictionary)
            {
                context.AdditionalResponseParameters.Add(property.Key, property.Value);
            }

            return Task.FromResult<object>(null);
        }

        public override Task
ValidateClientAuthentication(OAuthValidateClientAuthenticationContext context)
        {
            if (context.ClientId == null)
            {
                context.Validated();
            }

            return Task.FromResult<object>(null);
        }

        public override Task ValidateClientRedirectUri(OAuthValidateClientRedirectUriContext
context)
        {
            if (context.ClientId != _publicClientId)
            {
                return Task.FromResult<object>(null);
            }

            var expectedRootUri = new Uri(context.Request.Uri, "/");

            if (expectedRootUri.AbsoluteUri == context.RedirectUri)
            {
                context.Validated();
            }

            return Task.FromResult<object>(null);
        }

```

```

    }

    /// <summary>
    /// Создает набор параметров, хранимых в сессии аутентификации.
    /// </summary>
    /// <param name="userName">Имя пользователя.</param>
    public static AuthenticationProperties CreateProperties(string userName)
    {
        IDictionary<string, string> data = new Dictionary<string, string>
        {
            { "userName", userName }
        };

        return new AuthenticationProperties(data);
    }
}

```