**СОДЕРЖАНИЕ**

Введение…………………………………………………………………………….3

1 Общие сведения и структура предприятия «Фабрика инноваций и

решений»……………………………………………………………………………4

1.1 История предприятия «Фабрика инноваций и решений»….…………4

1.2 Охрана труда на предприятии…………………………………………..6

1.3 Политика информационной безопасности предприятия……………...8

1.4 Используемое оборудование...………………………………………….10

2 Существующие программы для автоматизации автобусных перевозок……...11

2.1 Программный комплекс *Bus Commander*……………………………....11

2.2 Программное обеспечение *Transport Manager*………………………...12

2.3 Облачный сервис *Octobus*……………………………………………….13

2.4 Программное обеспечение КВЦ-сервис………………………………..14

2.5 Мобильное приложение *BusTicketPro*………………………………….15

2.6 Программное обеспечение *Biletion Office*……………………………...15

2.7 Мобильное приложение Авибус………………………………………..17

2.8 Программный комплекс Е-Автовокзал…………………………………18

3 Этапы проектирования автоматизированной системы автобусных

перевезок……………………………………………………………………………20

3.1 Структура разработанного приложения….……………….....................20

3.2 Результаты тестирования пользовательского интерфейса..…………..30

Заключение………………………………………………………………………….36

Список использованных источников……………………………………………...37

Приложение А Должностная инструкция инженера-программиста…………....38

Приложение Б Листинг программы..…………………...…………...…………….41

**ВВЕДЕНИЕ**

Важнейшим этапом подготовки будущего специалиста является практика, позволяющая сформировать необходимые практические навыки на основе полученных теоретических знаний.

Основными целями практики являются:

– ознакомление с предприятиями и условиями труда в выбранной профессии;

– получение практических навыков работы в должности инженера-программиста;

– освоение в реальных условиях предприятия принципов организации и управления производством, анализа экономических показателей, проводимых мероприятий для повышения качества разрабатываемого программного обеспечения при снижении сроков его создания;

– закрепление полученных в процессе обучения знаний и навыков в области разработки программного обеспечения и использования современных информационных систем и технологий в проектировании и производстве;

– проверка возможности самостоятельной работы в условиях конкретной организации.

Во время прохождения практики студенты находятся в условиях реальной работы, что позволяет им получить дополнительные навыки в профессиональной деятельности, получить новые знания и опыт.

Теоретическое обучение создает основу знаний, формирует методику самостоятельного обучения и саморазвития. Выполнение заданий в процессе обучения формирует базовые навыки в области информационных технологий, знакомит со всем многообразием отрасли.

Практическая подготовка должна обеспечить студента профессиональными знаниями и умениями, которые должны охватывать всю будущую деятельность специалиста.

Таким образом, теоретические знания студенты получают во время выполнения заданий на лабораторных, практических и семинарских занятиях, а также во время работы над курсовыми проектами. Практическая подготовка осуществляется в ходе практик.

Целью преддипломной практики является проектирование автоматизированной системы для формирования отчетов автобусных перевозок. Данная система предназначена для удобного формирования отчетов по различным категориям записей и имеет интуитивно понятный пользовательский интерфейс.

**1 ОБЩИЕ СВЕДЕНИЯ И СТРУКТУРА ПРЕДПРИЯТИЯ**

**«ФАБРИКА ИННОВАЦИЙ И РЕШЕНИЙ»**

**1.1 История предприятия «Фабрика инноваций и решений»**

*Innowise Group* основана в 2007 году и уже более 13 лет занимается полным циклом разработки программного обеспечения. За это время были реализованы более 400 проектов в разных отраслях для клиентов из США, Канады, Великобритании, Германии, Франции, Италии, Японии и других стран.

*Innowise Group* работает над предоставлением программных решений и консалтинговых услуг предприятиям по всему миру, а также помощи клиентам в создании инновационных и технологических продуктов в различных областях.

Команда *Innowise Group* разделена на несколько отделов и структурных подразделений, ответственных за определенные направления деятельности компании.

Непрерывное сотрудничество между всеми ними на ежедневной основе помогает им достигать краткосрочных целей и стратегических целей.

*Innowise Group* ориентируемся на повышение эффективности бизнеса клиентов за счет следующих своевременных инноваций: автоматизация бизнес-процессов, системная интеграция, разработка специализированных баз данных, реинжиниринг существующих ИТ-систем, предоставление *SaaS* – *Software* *as* *a* *Service*.

Компания расширяет свое присутствие в областных центрах: в 2020 году открыты два новых офиса, и сейчас их центры разработки находятся в Витебске, Гомеле, Минске и Могилеве. Сегодня *Innowise Group* объединяет более 400 сотрудников и продолжает интенсивно расти, приглашая в свою команду как опытных профессионалов, так и молодых специалистов.

Среди основных направлений деятельности компании можно выделить следующие:

­– комплексные инфраструктурные решения и проекты по системной интеграции;

– полный цикл создания и сопровождения программного обеспечения (ПО) на основе решений, продуктов и технологий *IBM*, *Oracle*, *Microsoft*, *HPE* и других вендоров, а также *OpenSource* продуктов;

– центр обработки данных: «облачные» услуги по моделям *IaaS*, *PaaS*, *SaaS*;

– информационно-аналитические системы, комплексные системы анализа и управления эффективностью бизнеса, системы планирования/бюджетирования с возможностями консолидации и ситуационного моделирования на базе *BI*-платформ *IBM* *Cognos*, *Analytics*, *IBM* *Planning* *Analytics*, *SAP* *Business* *Objects* с использованием *SAP* *Fiori*, *Oracle* *BI*, *Oracle* *Hyperion* *BPC* и др.;

– системы анализа неструктурированной информации и связей в социальных сетях и системы анализа эмоциональной окраски текста по отзывам клиентов на базе *SAP Lumira*;

– системы статистического анализа и прогнозирования, системы проведения социологических опросов на базе *IBM SPSS, IBM Watson, SAP Predictive Analytics*, открытых технологий *Python* и *R Studio*;

– разработка и сопровождение систем и приложений на платформе *Mainframe*;

– разработка систем обработки транзакций;

– системы управления корпоративными мобильными устройствами (*Mobile Device Management*);

– системы управления ИТ-инфраструктурой (*OSS/BSS*), сервисами (*ITIL/ITSM*), обеспечение информационной безопасности, системы управления активами предприятия (*EAM*) на базе *IBM Cloud & Smarter Infrastructure (IBM Tivoli, IBM Maximo)*;

– системы информационной безопасности сетей, серверов, рабочих станций, бизнес-приложений, данных;

– системы управления доступом;

– системы управления соответствием стандартам и управления рисками;

* проектирование, разработка и сопровождение приложений для платформы *Java Enterprise Edition, Microsoft SharePoint/.Net*;
* приложений для извлечения, преобразования, загрузки массивов данных с использованием *ETL*-технологий;
* приложений в среде «облачных» вычислений;
* приложений для работы с *BigData* и *SemanticWeb*;
* приложений для мобильных устройств на различных платформах (*Android, iOS, WindowsPhone/Mobile*);
* приложений с использованием концепции *SPA* и сервис-ориентированной архитектуры (*SOA*);
* приложений на языке *ABAP* для систем *SAP*;
* приложений на базе системы коллективной работы *IBMNotes/Domino*;
* продуктов Интернет и *e*-*business* (информационных порталов, *Web*-сайтов, систем ведения электронных платежей и др.);
* решений для обмена данными между корпоративными системами, включая унаследованные, *SAP* и *Web*-приложения (на базе *IBM* *WebsphereMessageBroker*);
* создание и развитие интегрированных систем управления предприятием на базе решений *SAP*;
* миграция программных продуктов между разнообразными платформами и средами;
* проектирование и администрирование баз данных (*DB2, Oracle, MS SQL Server*);
* разработка и внедрение *BI*-решений по отчетности и анализу данных;
* разработка облачных сервисов по управлению НСИ (*MDM, MDG)*;
* анализ качества данных в справочниках (*ISO* 8000/22745);
* построение систем управления НСИ (*SAP MDG, MDM/BPM/Webdynpro*);
* построение онтологий и средств структуризации данных (*RDF/OWL*);
* преобразование данных НСИ к формату *SemanticWeb* и организация доступа (*SPARQL*);
* банковские технологии [1].

**1.2 Охрана труда на предприятии**

Охрана труда – система обеспечения безопасности жизни и здоровья, работающих в процессе трудовой деятельности, включающая правовые, социально-экономические, организационные, технические, санитарно-гигиенические, психофизиологические, лечебно-профилактические, реабилитационные, и иные мероприятия, и средства [2].

Руководством *Innowise Group* уделяется большое внимание улучшению эргономики рабочих мест, обеспечению гигиены и совершенствованию организации труда, регламентации режимов труда и отдыха.

Вся деятельность в области охраны труда в *Innowise Group* регламентирована действующим законодательством Республики Беларусь, санитарными нормами и правилами, гигиеническими нормативами, предписаниями надзорных органов.

К комплексу мероприятий в области охраны труда, осуществляемых в *Innowise Group* относятся:

– проведение периодических медицинских осмотров персонала;

– проведение производственного лабораторного контроля за условиями труда на рабочих местах;

– разработка инструкций по охране труда и ознакомление с ними персонала;

– обеспечение средствами индивидуальной защиты работников;

– изменение технологических процессов с целью совершенствования их безопасности для здоровья работников;

– модернизация рабочих мест и технологического оборудования;

– создание безопасных условий труда.

Важным фактором создания безопасных условий труда является оптимизация организации рабочих мест. При правильной организации рабочего места производительность труда инженера возрастает с 8 до 20 процентов.

Основным рабочим местом инженера-программиста является стол для выполнения машинописных работ. Основная поза при выполнении работы – вынужденная, сидячая.

Рабочее место для выполнения работ в положении сидя организуется в соответствии с ГОСТ 12.2.032–78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования» [3].

Требования к организации работы при использовании персонального компьютера и организационных средств определяются СанПиН 9–131 РБ 2000 «Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работы» [4].

Согласно ГОСТ 12.2.032–78 конструкция рабочего места и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия:

– оптимальное размещение оборудования, входящего в состав рабочего места;

– достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения;

– уровень акустического шума не должен превышать допустимого значения.

Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Помещения для работы программиста должны иметь естественное и искусственное освещение.

Площадь на одно рабочее место с видео-дисплейным терминалом (ВДТ) и ПЭВМ для взрослых пользователей должна составлять не менее , а объем не менее .

Искусственное освещение в помещениях эксплуатации ВДТ и ПЭВМ должно осуществляться системой общего равномерного освещения. В административно-общественных помещениях, в случаях преимущественной работы с документами, допускается применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк.

В качестве источников света при искусственном освещении необходимо использовать преимущественно люминесцентные лампы.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей (размер ВДТ и ПЭВМ, клавиатуры и др.), характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики.

Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии не менее чем 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Помимо требований к организации рабочего места СанПиН 9–131 РБ 2000 устанавливает требования к микроклимату рабочей зоны: влажности, температуре, скорости потока воздуха и пр.

Техника безопасности – это система организационных мероприятий и технических средств, предотвращающих воздействие на работающих опасных производственных факторов.

Для соблюдения техники безопасности в своей деятельности инженер-программист должен руководствоваться инструкцией по охране труда для программиста при выполнении работ с применением ПЭВМ и ВДТ.

**1.3 Политика информационной безопасности предприятия**

Высшее руководство организации, учитывая современные направления развития систем менеджмента, и факторы, влияющие на имидж и повышение конкурентоспособности, осознавая необходимость обеспечения информационной безопасности для успешного ведения бизнеса и повышения зрелости корпоративного управления, определило следующие цели в области информационной безопасности при проектировании, разработке, производстве, сопровождении программного обеспечения и автоматизированных информационных систем, а также при выполнении процессов внутренней деятельности:

– обеспечить соответствие законодательству, обязательным, контрактным (договорным) требованиям в области информационной безопасности;

– обеспечить защиту конфиденциальности, целостности и доступности информационных активов, используемых при реализации процессов, активов инженерной и технологической инфраструктур на основе оценки и обработки рисков информационной безопасности;

– обеспечить улучшение качества бизнес-планирования и инвестиций на основе учета информационных рисков, связанных с реализацией планов организации;

– минимизировать возможный ущерб от инцидентов информационной безопасности путём оперативного реагирования на их появление и ликвидации их последствий;

– обеспечить защиту прав и свобод человека при обработке его персональных данных, в том числе права на неприкосновенность частной жизни, личную и семейную тайну, и унификацию порядка обработки персональных данных в организации с требованиями международного права и законодательства стран, где работает организация;

– обеспечить постоянное улучшение системы менеджмента информационной безопасности (СМИБ).

Для достижения поставленных целей высшее руководство намерено обеспечить:

– результативное функционирование и постоянное совершенствование СМИБ в соответствии с международными требованиями, изложенными в стандарте СТБ *ISO/IEC* 27001, используя подход, основанный на управлении рисками информационной безопасности, и поддерживая допустимый уровень рисков в заданных пределах;

– применение риск-ориентированного мышления для достижения оптимального баланса при принятии решений на этапах планирования, выполнения, мониторинга, постоянного улучшения и повышение результативности СМИБ;

– необходимые ресурсы для осуществления менеджмента риска и применения результативных средств защиты информационных активов для снижения рисков;

– снижение рисков информационной безопасности за счет применения организацией передовых информационных технологий, используемых в них методов и средств обработки, хранения и передачи информации, встроенных средств защиты;

– обучение персонала в области информационной безопасности и достаточную компетентность, соответствующую установленной ответственности;

– необходимые условия для заинтересованности персонала в соблюдении установленных мер по обеспечению информационной безопасности и противодействию потенциальным угрозам.

Политика информационной безопасности обязательна для использования всеми сотрудниками, а также лицами, работающими с информацией организации в рамках заключенных контрактов.

Каждому сотруднику организации предоставляется доступ к активам ИТ организации, используемым при реализации бизнес-процессов и процессов внутренней деятельности организации.

Для получения доступа к активам сотрудник должен быть авторизован в ИС организации.

При авторизации сотруднику должен присваиваться уникальный идентификатор (как правило, совпадающий с его фамилией) и предоставляться права доступа к активам.

Права доступа сотрудника к активам должны предоставляться в объеме, необходимом сотруднику для выполнения его функциональных и должностных обязанностей.

Каждому сотруднику организации предоставляются по умолчанию минимальные права доступа к ИТ-активам (ресурсам технологической инфраструктуры и приложениям), которые необходимы ему для коммуникационного взаимодействия с другими сотрудниками, службами и подразделениями организации.

Должностная инструкция инженера-программиста представлена в приложении А.

**1.4 Используемое оборудование**

Рабочие кабинеты *Innowise Group* оснащены современным оборудованием, а также всеми сопутствующими необходимыми аксессуарами. Типичное рабочее место инженера-программиста состоит из следующих частей:

– персональный компьютер на базе процессора *Intel Core* (*i5*, *i7* или *i9*);

– монитор с диагональю 19-21 дюйм;

– клавиатура, мышь, наушники;

– доступ к лицензированному программному обеспечению;

– доступ к внутренним данным предприятия.

Кроме этого, офис оснащён проводной и беспроводной (*Wi-Fi*) сетью, объединяющей все компьютеры организации. Данные сети, также, предоставляют доступ к сети Интернет. В целом, можно отметить, что материальная база *Innowise Group* находится на высоком уровне.

**2 СУЩЕСТВУЮЩИЕ ПРОГРАММЫ ДЛЯ АВТОМАТИЗАЦИИ**

**АВТОБУСНЫХ ПЕРЕВОЗОК**

**2.1** **Программный комплекс *Bus Commander***

*Bus Commander –*система автоматизации пассажирских перевозок. Решение для обеспечения централизованного управления структурой. Объединяет в себе все интернет-ресурсы компании, диспетчерскую службу, агентов, водителей и пассажиров.

Возможности *Bus Commander* включают в себя:

**–**мобильное приложение для водителей для взаимодействия с диспетчером;

**–**мобильное приложение для пассажиров на *iPhone*, *Android*, *Windows* *Phone*;

– наличие статуса заявки, указываемого водителем, благодаря которому диспетчеры всегда знают актуальные данные о явке/неявке пассажиров;

**–**использование колл-центра для увеличения количества принятых звонков;

**–**СМС/*push*/*e*-*mail* уведомления, благодаря которым осуществялется информирование пассажиров о данных по рейсу, освободившихся местах, статусе заказа;

**–**система внутренних сообщений для коммуникации с администрацией;

– встроенные инструменты для формирования отчетности, позволяющие создавать отчеты в режиме реального времени или за выбранный период;

– контроль работы сотрудников;

– планирование расписаний;

– интеграция системы с платежными системами.

В мобильном приложении для водителей доступна вся необходимая информация, обновляющаяся в режиме реального времени: точки посадки и высадки, тарифы, контактные данные пассажиров, время отправления, количество забронированных мест, комментарии диспетчеров. Также из приложения можно осуществлять звонки.

В мобильном приложении для пассажиров бронирование билетов на рейсы осуществляется в несколько кликов. Данные о местоположении транспорта перед посадкой в автобус. Возможность оставления отзывов об уровне обслуживания или качестве предоставляемых услуг.

Стоимость использования системы рассчитывается индивидуально и зависит от размера пассажиропотока. Лицензии не имеют ограничений в количестве транспорта и рабочих мест операторов. Абонентская плата включает в себя техническую поддержку и обновления системы. Предусмотрен также демо-доступ к *Bus Commander* в течение 30 дней.

Недостатки *Bus Commander*:

– высокая стоимость обслуживания приложения;

– плохое сопровождение приложения в процессе использования;

– неудобный интерфейс [5].

**2.2** **Программное обеспечение *Transport Manager***

*Transport Manager –* программное обеспечение для управления пассажирскими перевозками. Сервис позволяет решать четыре задачи – оптимизировать бизнес, внедрить услугу бронирования билетов онлайн, автоматизировать планирование маршрутов, обеспечить контроль над автотранспортом.

Система предназначена компаниям перевозчикам, которым требуется единый сервис для управления бизнес-процессами и взаимодействия с клиентами. В программе могут работать руководители, диспетчеры и операторы, решая определённые задачи – от регистрации пассажиров и составления маршрутов до контроля загруженности рейсов и корректировки рейсов в режиме реального времени.

Преимущества *Transport Manager* включают в себя:

**–**система состоит из моделей управления автобусами, операторами, направлениями, водителями, расписаниями;

**–**автоматизация пассажирских перевозок;

**–**создание рейсов вручную, а также возможность привязки автобусов, водителей, схема посадки пассажиров;

**–**автоматическое составление расписаний на выбранное количество дней;

– наличие приложения для водителей, которое синхронизировано с системой, благодаря чему водитель получает все данные в режиме актуального времени, и позволяет отслеживать все предстоящие рейсы, регистрировать пассажиров на рейсе, видеть место посадки и высадки пассажиров, отмечать явку/неявку пассажиров;

– наличие модуля онлайн-бронирования, который адаптирован под мобильные устройства и позволяет клиентам бронировать билеты без помощи операторов;

– быстрое бронирование, которое заключается в том, что клиенты сами могут бронировать билеты на конкретный день и место, выбирать место посадки и высадки;

– ведение многофункциональной клиентской базы, которую можно использовать для сбора статистики, проведения рассылок и осуществления новых продаж;

– *GPS*-контроль, позволяющий увидеть местоположение автобуса водителю, пассажирам, оператору и руководителю и работающий в мобильном приложении, а также в системе на персональном компьютере;

– быстрая генерация отчетов за выбранный промежуток времени.

Недостатки *Transport Manager*:

– высокая стоимость обслуживания приложения;

– отсутствие возможности клиентам оставлять отзывы;

– неудобный интерфейс [6].

**2.3 Облачный сервис *Octobus***

*Octobus –* облачный сервис для управления автобусными перевозками. Используется как автостанциями и автобусными перевозчиками, так и агентами.

Возможности *Octobus* включают в себя:

– организация Интернет продаж билетов за несколько минут, которая интегрирована с платежными системами;

– ведение клиентской базы;

– сегментация пассажиров по различным признакам, выполнение СМС-рассылок;

– наличие инструментов для создания программ лояльности и формирования гибких цен;

– построение колл-центра по обслуживанию пассажиров и продаже билетов;

– контроль работы операторов и анализ статистики звонков;

– организация точек продаж при помощи планшета или ноутбука;

– мобильное приложение для пассажиров, позволяющее покупать билеты с любого мобильного устройства;

– мобильное приложение для диспетчеров и водителей;

– построение агентской сети, при помощи которой можно пользоваться услугами подключенных агентов, контролировать продажи и задолженности, заключать соглашения и повышать уровень продаж;

– управление заказами, рейсами, билетами и инвентарем;

– интеграции с внешними сервисами;

– продажа текущих и предварительных билетов;

– продажа билетов в обе стороны, с пересадками, мультимодальных и многосегментных билетов;

– мобильное приложение для пассажиров;

– подключение систем бронирования перевозчиков или дистрибьюторов для продажи в «одном окне»;

– продажа, допродажа, возврат, отмена и переоформление билетов;

– формирование отчетов о продажах в различных формах и разрезах;

– интеграция отчетности с распространенными бухгалтерскими программами.

Недостатки *Octobus*:

– высокая стоимость обслуживания приложения;

– неудобный интерфейс [7].

**2.4** **Программное обеспечение КВЦ-сервис**

КВЦ-сервис – система автоматизированной продажи билетов на автовокзалах от «КВЦ-сервис». Система для автоматизации и управления работой автовокзалов и их сетей. Система применяется как на малых автостанциях, так и на крупных автовокзалах.

Преимущества Системы автоматизированной продажи билетов на автовокзалах от «КВЦ-сервис»:

– технология штрихового кодирования для пассажиров со льготами;

– межрегиональная справочно-информационная система для получения справок через Интернет с другого автовокзала, работающего в системе;

– продажа текущих и предварительных билетов;

– возможность продажы международных билетов, с открытой датой, в обратном направлении, с других автовокзалов, с любой остановки установленного маршрута;

– допродажа билетов на рейс с уже распечатанной пассажирской ведомостью;

– печать дополнительной посадочной ведомости;

– аннулирование испорченных или ошибочно пробитых билетов;

– возврат билетов без/с удержанием комиссионного сбора;

– три варианта работы кассового пункта: с постоянным соединением с сервером автовокзала, без связи с сервером автовокзала и через Интернет;

– автоматизированное рабочее место администратора, позволяющее создавать новые маршруты, изменять расписание движения транспорта, изменять тарифы, формировать отчетность и просматривать аналитические данные;

– ведение диспетчерского контроля, позволяющий проверять оформление всех рейсов смены;

– контроль прохождения медицинского осмотра водителями и технического осмотра автобуса;

– формирование отчетов о выручке кассиров в разрезе автопредприятия, статуса рейса или выбранного периода времени и выручке кассиров от предварительной продажи билетов.

Недостатки КВЦ-сервис:

– высокая стоимость обслуживания приложения;

– устарело;

– отсутствует система отслеживания маршрута;

– неудобный интерфейс [8].

**2.5 Мобильное приложение *BusTicketPro***

*BusTicketPro* – облачная система учета пассажирских рейсов и продажи автобусных билетов. Размещается в дата-центре, благодаря чему возможно использование системы через сеть Интернет.

Продукт *BusTicketPro* был разработан европейской компанией *SWD* *Factory*, специализирующейся на создании программных решений для оптимизации бизнес-процессов. По словам разработчиков, их главная цель – «приблизить услугу автотранспортной перевозки ближе к пассажиру, сделать процесс покупки билетов максимально простым и удобным». На сегодняшний день *BusTicketPro* используется сотнями успешных компаний по всей Европе.

Возможности *BusTicketPro* включают в себя:

– круглосуточный доступ к данным;

– возможность быстрого масштабирования системы в случае роста бизнеса;

– высокий уровень защиты конфиденциальных данных и их безопасное хранение;

– наличие восьми основных пользователей системы: администраторы, диспетчеры, агентства по продаже билетов, перевозчики, справочные службы, кассиры, клиенты, руководители подразделений;

– продажа билетов в кассе, в Интернет, в агентствах;

– ведение и учет тарифов, расписаний, остановок, рейсов, схем и марок автобусов;

– управление скидками и программами лояльности;

– интеграция с фискальным принтером и кассовым аппаратом;

– контроль фактического времени отправления и прибытия автобусов;

– экспорт данных в бухгалтерскую систему;

– наличие мультивалютности;

– формирование широкого спектра отчетов;

– информационная система, содержащая информацию об актуальности рейсов, наличии свободных мест и т.д.;

– возможность добавления новых пользователей в систему, например, сотрудников, агентств или филиалов.

Недостатки *BusTicketPro*:

– высокая стоимость обслуживания приложения;

– отсутствие отслеживания статуса заказа;

– неудобный интерфейс [9].

**2.6** **Программное обеспечение *Biletion Office***

*Biletion Office –*программа автоматизации и учета пассажирских перевозок

международного и междугороднего сообщения. Программа *Biletion* *Office* предоставляет руководителю полный арсенал отчетов и историй заявок, чтобы иметь полную картину о работе компании в любое время и в любом месте. Кроме того, с использованием программы сводятся к минимуму случаи воровства и мошенничества как среди сотрудников компании, так и среди пассажиров.

Возможности *Biletion Office*:

– ведение клиентской базы, которая интегрирована со всеми остальными модулями программы;

– автоматическое создание рейсов;

– быстрое создание новой заявки, для создания которой требуется буквально несколько минут;

– мобильное приложение для водителей *Biletion Mobile*, в котором содержится актуальная информация обо всех рейсах и пассажирах, а также ведется коммуникация между диспетчером и водителем;

– мобильное приложение для клиентов на *Android*, *iOS* и *Windows* *Phone*, в котром можно бронировать билеты, отслеживать информацию о рейсах, заказывать обратный звонок диспетчера, отменять или оплачивать бронь, просматривать свою историю заявок;

– майт для онлайн-бронирования;

– СМС-оповещение;

– автоматическое информирование звонком, представляющий собой набор голосовых оповещений;

– *GPS*-мониторинг, который заключается в отслеживании местоположения автобуса водителем, пассажиром, диспетчером и руководителям компании;

– онлайн отчетность, позволяющая быстрое формирование отчетов за любой выбранный период времени.

Единая клиентская база компании. Карточка каждого клиента содержит такие данные: ФИО, паспортные данные, адрес, телефон, электронная почта, пароль в личный кабинет, статистика и история поездок.

Расписание регулярных рейсов может создаваться автоматически, а также вручную. К рейсам привязываются автобусы, водители, способ сбор пассажиров и др. параметры.

Сайт для онлайн-бронирования позволяет клиентам самостоятельно выбирать и оплачивать билеты. Все поступающие заявки и отказы попадают в систему в режиме реального времени. Модуль также может быть установлен на уже существующий сайт.

СМС-оповещение включает в себя возможности по автоинформированию пассажира о рейсе, рассылке рекламных сообщений, информированию об изменениях в расписании или задержке рейса.

Недостатки *Biletion Office*:

– высокая стоимость обслуживания приложения;

– плохая масштабируемость [10].

**2.7** **Мобильное приложение Авибус**

Авибус – мобильная система для организации продажи билетов, решения задач автоматизации и управления деятельностью автовокзалов, автостанций и их сетей.

ВозможностиАвибус:

– объединение нескольких автовокзалов в единую сеть. Продажа билетов с любого автовокзала сети;

– установка и ведение тарифов и тарифных сеток;

– контроль открытия и закрытия кассовых смен;

– предварительная и текущая продажа билетов, продажа на транзитные места;

– бронирование, возврат и выкуп билетов;

– возврат средств за неоконченный путь;

– оплата наличными, банковскими картами, с сайта при помощи банковских карт, через терминал самообслуживания;

– поддержка эквайринговых терминалов и фискального оборудования;

– учет фискальных регистраторов;

– организация базы данных (централизованной или распределенной);

– регистрация прибытия, отправки и отмены рейсов;

– проведение операций с перевозчиками: транспортные средства, договоры, водители, схемы мест и т.д.;

– регистрация поломки автобуса;

– наличие мобильного приложения на *Android* для операторов автостанций и контролеров;

– управление квотами мест (исключение из продажи определенного количества билетов);

– управление транзитными местами и бронью;

– распределение доходов от неиспользованных билетов;

– расчеты с перевозчиками по договорам;

– автоматическое создание рейсов при помощи регламентного задания;

– создание дополнительных рейсов по маршруту;

– создание маршрутов, остановок и их привязка к карте;

– редактирование рейсов;

– посадка и удаление пассажиров;

– печать посадочной ведомости;

– голосовой автоинформатор, позволяющий воспроизводить звуковые сообщения;

– система разграничения прав доступа;

– формирование отчетов по диспетчеризации, продажам, итоговые списки документов.

Стоимость системы Авибус рассчитывается индивидуально. Она зависит от выбранных объектов и указанного количества пользователей для каждого из программных продуктов. Техническое сопровождение, подписка на обновления и доработки системы под индивидуальные особенности тарифицируются отдельно.

Недостатки Авибус:

– неудобный интерфейс;

– отсутствие возможности отслеживать состояние маршрута в режиме онлайн;

– плохая масштабируемость [11].

**2.8 Программный комплекс Е-Автовокзал**

Е-Автовокзал – программный комплекс для управления автовокзалами. Автоматизирует пассажирские перевозки на всех этапах.

ВозможностиЕ-Автовокзал:

**–**продажа билетов при помощи фискальных регистраторов;

**–**высокая скорость ввода персональных данных пассажиров;

**–**ведение персонифицированного учета льготных категорий и быстрая идентификация льготников;

– продажа билетов через терминалы самообслуживания;

– продажа билетов со всех автовокзалов сети;

– организация продажи билетов через Интернет;

– возможность интеграции продажи билетов в уже существующий сайт;

– интеграция расписания автобусов в уже существующий сайт;

– модуль диспетчера позволяет регистрировать отправку и отмену рейсов, управлять транзитными местами, вводить дополнительные маршруты, печатать отчеты и ведомости, отмечать явку и неявку пассажиров;

– печать талонов для пассажиров, приобретающих билеты через Интернет;

– автоматизация процесса голосового оповещения пассажиров о прибытии и отправке автобусов;

– модуль справки, включающий в себя инструменты для регистрации персональных данных пассажиров, бронирования билетов, получения данных о свободных местах и расписании;

– автоматический контроль сроков бронирования;

– возможность организации передачи данных о расписании на сервер автовокзала или внешним организациям;

– вывод информации на табло и телевизоры;

– формирование тарифов;

– управление правами доступа пользователей;

– регистрация перевозчиков;

– оформление договоров с агентами и перевозчиками;

– построение и модификация отчетов;

– интеграция с 1С Бухгалтерией.

Базовая версия автоматизированной системы управления (АСУ) «Е-Автовокзал» предоставляется бесплатно и включает в себя модули кассы, диспетчера, администратора, справки, коммуникационный сервис и сервис отчетов. Предлагается также расширение базовой версии комплекса, расширения тарифицируются отдельно. Отдельно также оплачиваются услуги по сопровождению работы в АСУ «Е-Автовокзал».

Недостатки Е-Автовокзал:

– неудобный интерфейс;

– плохая масштабируемость [12].

На основавнии сравнения подтверждается актуальность разработки автоматизированной системы для формирования отчетов автобусных перевозок. В состав возможностей приложения входят следующие функции:

– возможность редактировать, добавлять и выводить информацию по автобусам, зарегистрированным в системе;

– возможность редактировать, добавлять и выводить информацию по водителям, зарегистрированным в системе;

– возможность редактировать, добавлять и выводить информацию по маршрутам, зарегистрированным в системе;

– возможность редактировать, добавлять и выводить информацию по заказчикам, зарегистрированным в системе;

– возможность редактировать, добавлять и выводить информацию по расходам по маршрутам;

– возможность экспортировать данные из всех таблиц приложения в документы для формирования соответствующих отчетов.

**3 ЭТАПЫ ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННОЙ**

**СИСТЕМЫ АВТОБУСНЫХ ПЕРЕВЕЗОК**

**3.1 Структура разработанного приложения**

***3.1.1*** В состав разработанного приложения входят такие функции как:

– постраничное разбиение данных, полученных пользователем из разработанной базы данных, для их последующего отображения;

– масштабирование главного окна приложения;

– возможность формирования различных отчетов по таблицам в зависимости от указанного шаблона;

– возможность редактировать, добавлять и выводить информацию по записям из таблиц разработанной базы данных, зарегистрированным в системе;

– экспорт данных из всех таблиц разработанной базы данных в *excel-*документы для формирования соответствующих отчетов.

Подробное описание программы приведено ниже.

Код приложения приведен в приложении Б.

***3.1.2*** Для автоматизированной системы для формирования отчетов разработана иерархия классов, которая описывает данную предметную область. Для каждой сущности создан соответствующий класс, а с помощью системы управления базами данных *Microsoft SQL Server* сгенерирована база данных *TransportationAccountig.* Диаграмма базы данных приведена на рисунке 3.1.

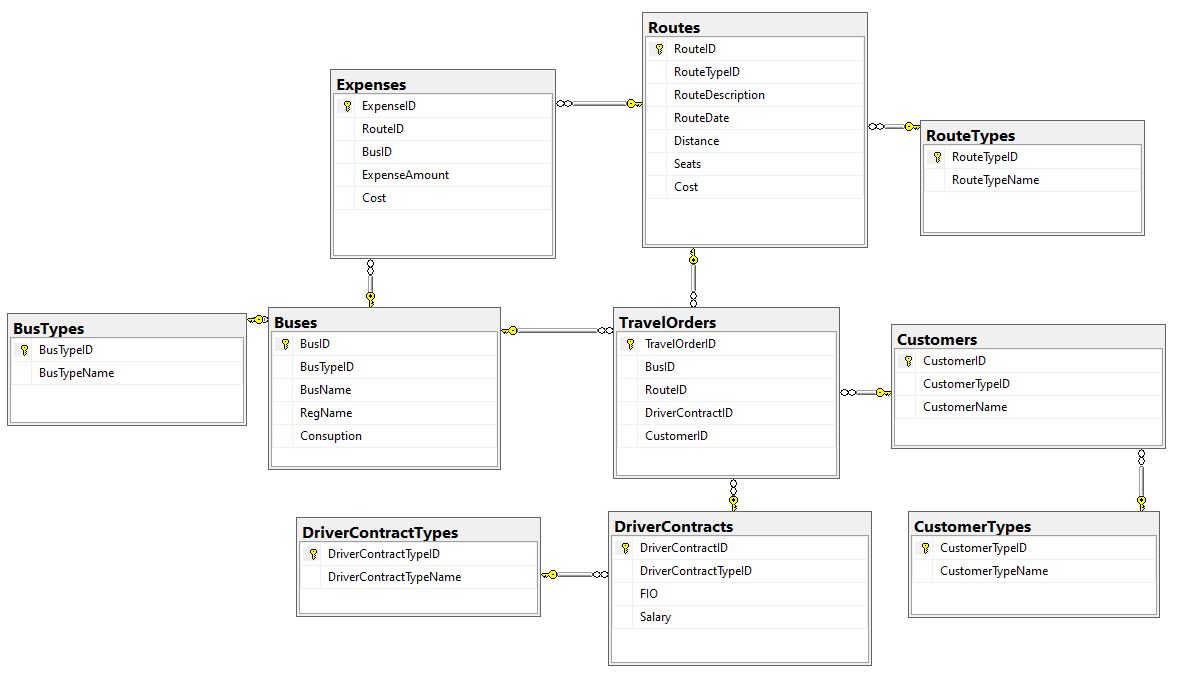


Рисунок 3.1 – Диаграмма базы данных «*TransportationAccountig*»

Связи типа «один-к-одному» имеют место, когда каждому атрибуту первой сущности соответствует только один атрибут второй сущности и наоборот, каждому атрибуту второй сущности соответствует только один атрибут первой сущности. Следует отметить, что такие сущности могут быть объединены в одну.

Связи типа «один-ко-многим» – это такие связи, когда каждому атрибуту одной сущности может соответствовать несколько атрибутов другой сущности, а каждому атрибуту второй сущности может соответствовать только один атрибут первой сущности. В такой связи первая сущность является главной, а вторая подчиненной.

Связи типа «многие-ко-многим» – это такие связи, когда каждому атрибуту первой сущности соответствует несколько атрибутов второй сущности и наоборот. Такие связи не могут непосредственно реализовываться в реляционной базе данных. Их можно реализовать путем введения дополнительного объекта «связка».

Первичный ключ – ключ, который используется для идентификации объекта, а составной, который использует несколько атрибутов [13].

Для сущности «Автобус» была спроектирована таблица «*Buses*» с атрибутами: «*BusID*», «*BusTypeID*», «*BusName*», «*RegName*», «*Consuption*». Данная таблица была приведена к первой нормальной с помощью замены вида автобуса на внешний ключ, связанный с таблицей, содержащей соответствующие записи. Описание атрибутов сущности «Автобус» приведено в таблице 3.1.

Таблица 3.1 – **Атрибуты, описывающие сущность «Автобус»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *BusID* | Уникальный инкрементируемый идентификатор для каждого автобуса. Является первичным ключом | *uniqueidentifier* |
| *BusTypeID* | Содержит неуникальное значение вида автобуса. Является внешним ключом | *uniqueidentifier* |
| *BusName* | Содержит название автобуса | *nvarchar(30)* |
| *RegName* | Содержит регистрационный номер автобуса | *nvarchar(12)* |
| *Consuption* | Содержит целочисленное значение расхода топлива на 100 км пути | *int* |

Для сущности «Вид автобуса» была спроектирована таблица «*BusTypes*» с атрибутами: «*BusTypeID*», «*BusTypeName*». Данная таблица находится в первой нормальной форме. Описание атрибутов таблицы приведено в таблице 3.2.

Таблица 3.2 – **Атрибуты, описывающие сущность «Вид автобуса»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *BusTypeID* | Уникальный инкрементируемый идентификатор для каждого вида автобуса. Является первичным ключом | *uniqueidentifier* |
| *BusTypeName* | Содержит описание вида автобуса по вместимости пассажиров | *nvarchar(50)* |

Для сущности «Заказчик» была спроектирована таблица «*Customers*» с атрибутами: «*CustomerID*», «*CustomerTypeID*», «*CustomerName*». Данная таблица была приведена к первой нормальной с помощью замены вида организации на внешний ключ, связанный с таблицей, содержащей соответствующие записи по каждой из категории. Описание атрибутов сущности «Заказчик» приведено в таблице 3.3.

Таблица 3.3 – **Атрибуты, описывающие сущность «Заказчик»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *CustomerID* | Уникальный инкрементируемый идентификатор для каждого заказчика. Является первичным ключом | *uniqueidentifier* |
| *CustomerTypeID* | Содержит неуникальное значение вида заказчика. Является внешним ключом | *uniqueidentifier* |
| *CustomerName* | Содержит наименование заказчика | *nvarchar(30)* |

Для сущности «Вид заказчика» была спроектирована таблица «*CustomerTypes*» с атрибутами: «*CustomerTypeID*», «*CustomerTypeName*». Данная таблица находится в первой нормальной форме. Описание атрибутов таблицы приведено в таблице 3.4.

Таблица 3.4 – **Атрибуты, описывающие сущность «Вид заказчика»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *CustomerTypeID* | Уникальный инкрементируемый идентификатор для каждого вида организации. Является первичным ключом | *uniqueidentifier* |
| *CustomerTypeName* | Содержит описание вида организации | *nvarchar(50)* |

Для сущности «Вид водительского договора» была спроектирована таблица «*DriverContractTypes*» с атрибутами: «*DriverContractTypeID*», «*DriverContractTypeName*». Данная таблица находится в первой нормальной форме. Описание атрибутов таблицы приведено в таблице 3.5.

Таблица 3.5 – **Атрибуты, описывающие сущность «Вид водительского договора»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *DriverContractTypeID* | Уникальный инкрементируемый идентификатор для каждого вида водительского договора. Является первичным ключом | *uniqueidentifier* |
| *DriverContractTypeName* | Содержит описание вида водительского договора | *nvarchar(50)* |

Для сущности «Водительский договор» была спроектирована таблица «*DriverContracts*» с атрибутами: «*DriverContractID*», «*DriverContractTypeID*», «*FIO*», «*Salary*». Данная таблица была приведена к первой нормальной с помощью замены вида водительского договора на внешний ключ, связанный с таблицей, содержащей соответствующие записи. Описание атрибутов сущности «Водительский договор» приведено в таблице 3.6.

Таблица 3.6 – **Атрибуты, описывающие сущность «Водительский**

**договор»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *DriverContractID* | Уникальный инкрементируемый идентификатор для каждого водительского договора. Является первичным ключом | *uniqueidentifier* |
| *DriverContractTypeID* | Содержит неуникальное значение вида водительского договора. Является внешним ключом | *uniqueidentifier* |
| *FIO* | Содержит фамилию, имя, отчество водителя | *nvarchar(50)* |
| *Salary* | Содержит значение зарплаты водителя | *money* |

Для сущности «Вид маршрута» была спроектирована таблица «*RouteTypes*» с атрибутами: «*RouteTypeID*», «*RouteTypeName*». Данная таблица находится в первой нормальной форме. Описание атрибутов таблицы приведено в таблице 3.7.

Таблица 3.7 – **Атрибуты, описывающие сущность «Вид маршрута»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *RouteTypeID* | Уникальный инкрементируемый идентификатор для каждого вида маршрута. Является первичным ключом | *uniqueidentifier* |
| *RouteTypeName* | Содержит описание вида маршрута | *nvarchar(50)* |

Для сущности «Маршрут» была спроектирована таблица «*Routes*» с атрибутами: «*RouteID*», «*RouteTypeID*», «*RouteDescription*», «*RouteDate*», «*Distance*», «*Seats*», «*Cost*». Данная таблица была приведена к первой нормальной с помощью замены вида маршрута на внешний ключ, связанный с таблицей, содержащей соответствующие записи. Описание атрибутов сущности «Маршрут» приведено в таблице 3.8.

Таблица 3.8 – **Атрибуты, описывающие сущность «Маршрут»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *RouteID* | Уникальный инкрементируемый идентификатор для каждого маршрута. Является первичным ключом | *uniqueidentifier* |
| *RouteTypeID* | Содержит неуникальное значение вида маршрута. Является внешним ключом | *uniqueidentifier* |
| *RouteDescription* | Содержит описание маршрута | *nvarchar(MAX)* |
| *RouteDate* | Содержит дату отправки по маршруту | *date* |
| *Distance* | Содержит расстояние по маршруту | *float* |
| *Seats* | Содержит доступное количество мест | *int* |
| *Cost* | Содержит стоимость билета | *money* |

Для сущности «Расходы» была спроектирована таблица «*Expenses*» с атрибутами: «*ExpenseID*», «*RouteID*», «*BusID*», «*ExpenseAmount*», «*Cost*». Данная таблица была приведена к первой нормальной с помощью замены автобуса и маршрута на внешние ключи, связанные с таблицами, содержащими соответствующие записи по каждой из категории. Описание атрибутов сущности «Расходы» приведено в таблице 3.9.

Таблица 3.9 – **Атрибуты, описывающие сущность «Расходы»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| *ExpenseID* | Уникальный инкрементируемый идентификатор для каждого расхода. Является первичным ключом | *uniqueidentifier* |
| *RouteID* | Содержит неуникальное значение маршрута. Является внешним ключом | *uniqueidentifier* |
| *BusID* | Содержит неуникальное значение автобуса. Является внешним ключом | *uniqueidentifier* |
| *ExpenseAmount* | Содержит общее значение расхода | *float* |
| *Cost* | Содержит стоимость топлива | *money* |

Для сущности «Путевой лист» была спроектирована таблица «*TravelOrders*» с атрибутами: «*TravelOrderID*», «*BusID*», «*RouteID*», «*DriverContractID*», «*CustomerID*». Данная таблица была приведена к первой нормальной с помощью замены автобуса, маршрута, водителя и заказчика на внешние ключи, связанные с таблицами, содержащими соответствующие записи по каждой из категории. Описание атрибутов сущности «Путевой лист» приведено в таблице 3.10.

Таблица 3.10 – **Атрибуты, описывающие сущность «Путевой лист»**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание домена** | **Тип данных** |
| 1 | 2 | 3 |
| *TravelOrderID* | Уникальный инкрементируемый идентификатор для каждого путевого листа. Является первичным ключом | *uniqueidentifier* |
| *BusID* | Содержит неуникальное значение автобуса. Является внешним ключом | *uniqueidentifier* |

Продолжение таблицы 3.10

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| *RouteID* | Содержит неуникальное значение маршрута. Является внешним ключом | *uniqueidentifier* |
| *DriverContractID* | Содержит неуникальное значение водителя. Является внешним ключом | *uniqueidentifier* |
| *CustomerID* | Содержит неуникальное значение заказчика. Является внешним ключом | *uniqueidentifier* |

***3.1.3*** Для реализации работы в приложении используется объектно-ориентированный подход. Объектно-ориентированное программирование (ООП) представляет собой способ программирования, который напоминает процесс человеческого мышления. ООП более структурировано, чем другие способы программирования и позволяет создавать модульные программы с представлением данных на определенном уровне абстракции. Основная цель ООП – это повышение эффективности разработки программ.

Весь окружающий нас мир состоит из объектов, предметов живой и неживой природы, которые представляются как единое целое, а отдельные части объектов образуют сложное взаимодействие друг с другом. При структурном подходе программист обычно разделяет (структурирует) описываемый объект на составные части, стараясь описать свойства отдельных частей, не вдаваясь в подробности взаимодействия между ними.

Базовым в ООП является понятие объекта. Объект имеет определенные свойства. Состояние объекта задается значениями его признаков. Объект «знает», как решать определенные задачи, т.е. располагает методами решения. Программа, написанная с использованием ООП, состоит из объектов, которые могут взаимодействовать между собой. Концепция ООП заключается в том, что каждый объект является экземпляром некоторого класса объектов [14].

Для описания предметной области реализованы классы, отображающие состояние той или иной сущности. Реализованы следующие классы:

– *City –* класс, характеризующий сущность «Город»;

*– Client –* класс, характеризующий сущность «Клиент»;

*– Project –* класс, характеризующий сущность «Проект»;

*– ProjectType –* класс, характеризующий сущность «Тип проекта»;

*– User –* класс, характеризующий сущность «пользовтаель»;

*– UserRole –* класс, характеризующий сущность «роль пользователя».

Диаграмма классов-сущностей предоставлена на рисунке 2.2.

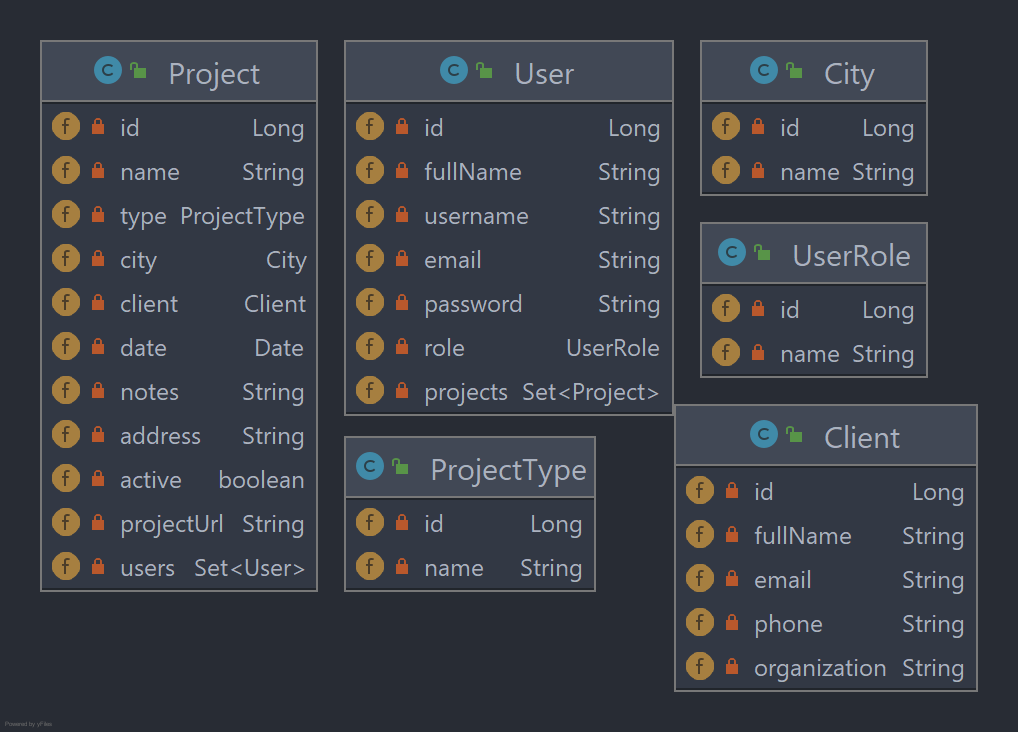


Рисунок 2.2 – Диаграмма классов-сущностей

Для доступа к базе данных используется фреймворк *Hibernate*. Для взаимодействия с базой данныз, необходимо классы-сущности пометить аннотацией @*Table,* а также создать наследуемые от *JpaRepository*<*T*, *T*> интерфейсы, представляющие собой репозитории. JpaRepository хранит в себе ряд готовых методов, которые взаимодействуют с сущностью. Например, методы: *findAll*, *findById*, *create*, *update*, *delete*. Тоесть все *CRUD* методы. Также, если это необходимо, можно определеить свои собственные методы, например, *findByEmail* у пользователя.

Список разработанных интерфейсов:

– *UserRepository* – интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *User*>;

– *UserRoleRepository* – интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *UserRole*>;

– *CityRepository*– интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *City*>;

– *ClientRepository*– интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *Client*>;

– *ProjectRepository*– интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *Project*>;

– *ProjectTypeRepository*– интерфейс для доступа к базе данных, наследующий *JpaRepository*<*Long*, *ProjectType*>;

В интерфейсе City*Repository* определены следующие собственные методы:

– *findByName*(*String name*) – метод, позволяющий получать город по определенному имени;

– *findAll*(*Pageable pageable*) – метод, позволяющий выбирать все записи о городах, помещенные в *Page*<>.

В интерфейсе *ClientRepository*определены следующие методы:

– *findByEmail*(*String name*) – метод, позволяющий получать клиента по его почте;

– *findAll*(*Pageable pageable*) – метод, позволяющий выбирать все записи о городах, помещенные в *Page*<>.

В интерфейсе *ProjectRepository*определены следующие методы:

– *findAll*(*Pageable pageable*) – метод, позволяющий выбирать все записи о городах, помещенные в *Page*<>;

– *findUserProjects*(*Long userId*) – метод, возвращающий проекты пользователя с определенным *userId*;

– *findProjectsBetweenTwoDates*(*Date startDate, Date endDate*) – метод, возвращающий проекты, которы находятся в диапазоне между двумя датами.

Для реализации серверной части используется *Spring Boot*. Контроллеры создаются для каждой из сущностей и реализуют все необходимые *CRUD* операции.

Контроллер *CityController*:

– *Method GET /api/admin/city/* – маршрут позволяющий получать страницу городов. В параметрах принимает Pageable объект, который хранит в себе *pageSize*, *page*, *sort*;

– *Method GET /api/admin/city/{id*} – маршрут позволяющий получать город по его *id*. В параметрах принимает *id* сущности, которую необходимо получить;

– *Method POST /api/admin/city*/ – маршрут позволяющий создавать город. В теле запроса принимает *json* объект *CityDto*.

– *Method PUT /api/admin/city/{id}* – маршрут позволяющий обновлять сущность по его *id*. В параметрах принимает *id* сущности, а в теле запроса *json* объект *CityDto* с обновленными полями.

– *Method DELETE /api/admin/city/{id}* – маршрут позволяющий удалять сущность из базы данных. В параметрах принимает *id* сущности.

Остальные контроллеры выполнены по такому же сценарию. Все контроллеры реализуют все *CRUD* операции для сущности.

Для каждого контроллера необходимо создать сервис, в котором будет идти взаимодействие с репозиторием. Сервисы вызываются в контроллере и получают объект *Dto*, который преобразуют в сущность базы данных и отправляют в репозиторий. Преобразование классов реализовано с помощью *Mapstruct*.

**3.2 Результат тестирования пользовательского интерфейса**

Для работы приложения необходимо скомпилировать библиотеки *Contracts.dll, EntitiesFeatures.dll, LoggerService.dll, Repository.dll,* а также наличие скомпилированной программы пользовательского интерфейса *Transportation-Accounting.exe*.

Верификация разработанного приложения заключается в проверке требований к функциям разработанной системы. Одним из требований к разрабатываемой системе является наличие графического пользовательского интерфейса.

После запуска приложения путем открытия исполняемого файла *Trans-portationAccounting.exe*. пользователю будет предоставлен доступ к главному меню. Пример внешнего вида графического интерфейса приведен на рисунке.

После отображения графического интерфейса пользователю предоставляется возможность работать с данными, хранящимися в подключенной базе данных. Для этого пользователю доступны следующие пункты меню в меню «Ввод данных»:

– Типы автобусов – пункт меню, позволяющий работать с данными по типам автобусов;

– Автобусы – пункт меню, позволяющий работать с данными по автобусам;

– Виды заказчиков – пункт меню, позволяющий работать с данными по видам заказчиков;

– Заказчики – пункт меню, позволяющий работать с данными по заказчикам;

– Виды контрактов водителей – пункт меню, позволяющий работать с данными по видам контрактов водителей;

– Водители – пункт меню, позволяющий работать с данными по водителям;

– Виды маршрутов – пункт меню, позволяющий работать с данными по видам маршрутов;

– Маршрут – пункт меню, позволяющий работать с данными по маршрутам;

– Путевые листы – пункт меню, позволяющий работать с данными по путевым листам;

– Учет топлива – пункт меню, позволяющий работать с данными по учету расходов топлива по маршруту.

Внешний вид окна меню «Вывод данных» изображен на рисунке 3.2.



Рисунок 3.2 – Внешний вид окна меню «Вывод данных»

Для вывода записей из таблиц разработанной базы данных пользователю необходимо выбрать меню «Вывод данных», после чего отобразиться подменю для выбора таблицы для вывода данных, в котором пользователю необходимо определиться, содержимое какой таблицы необходимо отобразить. Пример вывода всех автобусов, находящихся в системе изображен на рисунке 3.3.

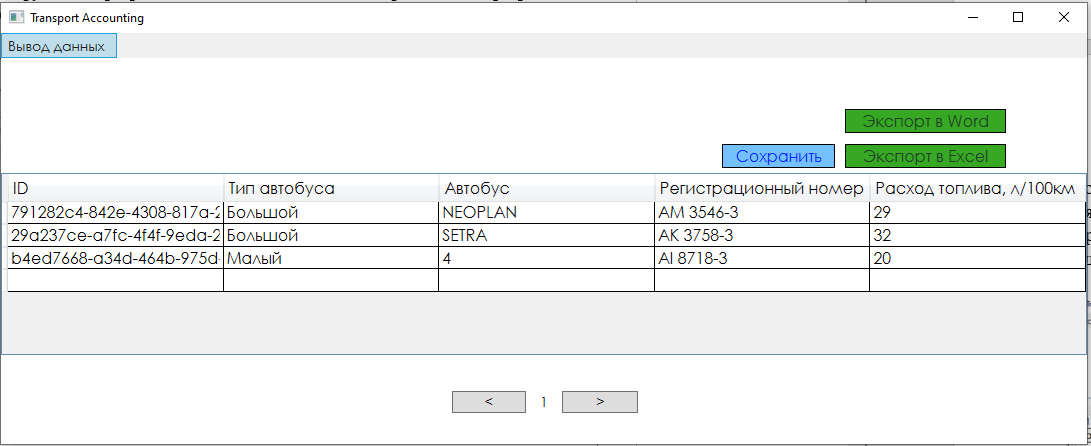


Рисунок 3.3 – Пример вывода всех записей из таблицы *Buses*

После отображения данных из какой-либо таблицы пользователю предо-

ставляется возможность редактирования, удаления и добавления записей в таблице. Для удаления пользователю необходимо выбрать запись для удаления и нажать на правую кнопку мыши (ПКМ), после чего отобразить окно для удаления записи. В окне для удаления отобразится информация об удаляемой записи с кнопками подтверждения и отмены выбора. Пример окна для удаления записи показан на рисунке 3.4.

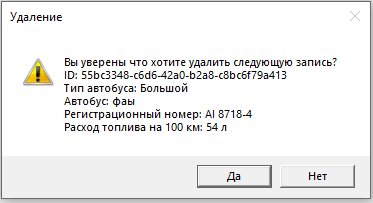


Рисунок 3.4 – Окно удаления записи

После успешного удаления записи пользователю отобразится соответствующее сообщение, а запись исчезнет из списка доступных записей. Пример сообщения об успешном удалении представлен на рисунке 3.5.

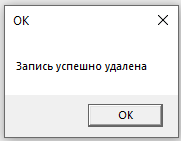


Рисунок 3.5 – Сообщение об успешном удалении

Вместе с возможностью удаления пользователю предоставляется доступ к добавлению новых записей и изменению уже существующих. Для того, чтобы добавить новую запись, пользователю необходимо дважды нажать на пустое поле под всеми записями, в результате чего появится возможность для ввода данных. Для редактирования существующих записей пользователю необходимо выбрать запись для редактирования из списка доступных и дважды нажать на поле ввода этой записи, после чего запись станет активна для внесения изменений. Для успешного сохранения изменений пользователю необходимо нажать кнопку «Сохранить». Если поля для ввода заполнены неподходящими значениями, то пользователю отобразится соответствующие сообщение с просьбой проверки данных ввода. Внешний вид сообщения о некорректном вводе показан на рисунке 3.6.

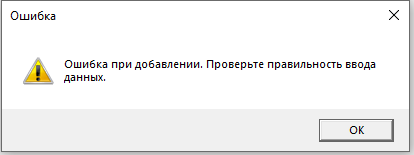


Рисунок 3.6 – Сообщение об ошибке при сохранении данных

Так как в приложении реализована система постраничного отображения данных таблиц, то пользователю доступна постраничная навигация с помощью кнопок в нижней части графического интерфейса. Путем нажатия на копки пользователь может переключать содержимое таблицы в соответствии с максимальным размером страницы. В случае, если записи закончились, а пользователь вновь пытается осуществить переход на следующую страницу, то ему отобразится сообщение о том, что все записи уже отображены. Пример сообщения об отображении всех записей показан на рисунке 3.7.

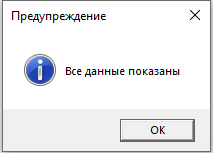


Рисунок 3.7 – Сообщение об отображении всех записей

Главной функцией разработанного приложения является экспортирование данных в *excel-*документы для последующего формирования отчетов. Для того, чтобы экспортировать данные, пользователю необходимо выбрать данные для экспорта путем отображения окна с ними из меню «Вывод данных». После чего, пользователю станет доступна функция экспорта путем нажатия на кнопку «Экспорт в *Excel*» После нажатия пользователю отобразится окно файловой системы, где он должен выбрать путь сохранения файла в формате *.xls* или *.xlsx.* Если путь к файлу не указан выведется соответствующее сообщение. Пример сообщения об ошибке показан на рисунке 3.8.

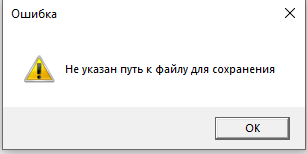
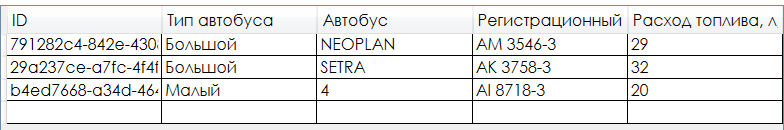


Рисунок 3.8 – Сообщение об ошибке при указании путь к файлу для сохранения

Для формирования отчетов по шаблонам пользователю необходимо выбрать таблицу, записи которох будут экспортироваться в *doc-*документы, путем отображения окна с ними из меню «Вывод данных». После выбора пользователю предоставляется возможность экспорта в *doc-*документы по нажатию на кнопку «Экспорт в *Word*», после нажатия на которую необходимо указать путь к файлу шаблона и путь к файлу сохранения.

В случае успешного экспорта данных в *excel-*документ пользователю отобразится содержимое файла, в который экспортируются данные. Результат сравнения данных, хранящихся в таблице *Buses* в базе данных, и содержимого *excel-*документа, в который эти данные экспортировались, показан на рисунке 3.9.



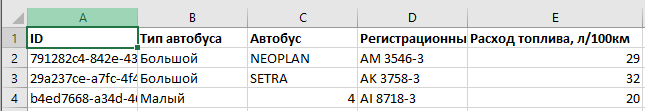


Рисунок 3.9 – Результат сравнения

Из рисунка 3.9 видно, что экспорт данных прошел успешно и нет никаких ошибок.

Результат формирования отчета по путевому листу из записей в таблице показан на рисунке 3.10.

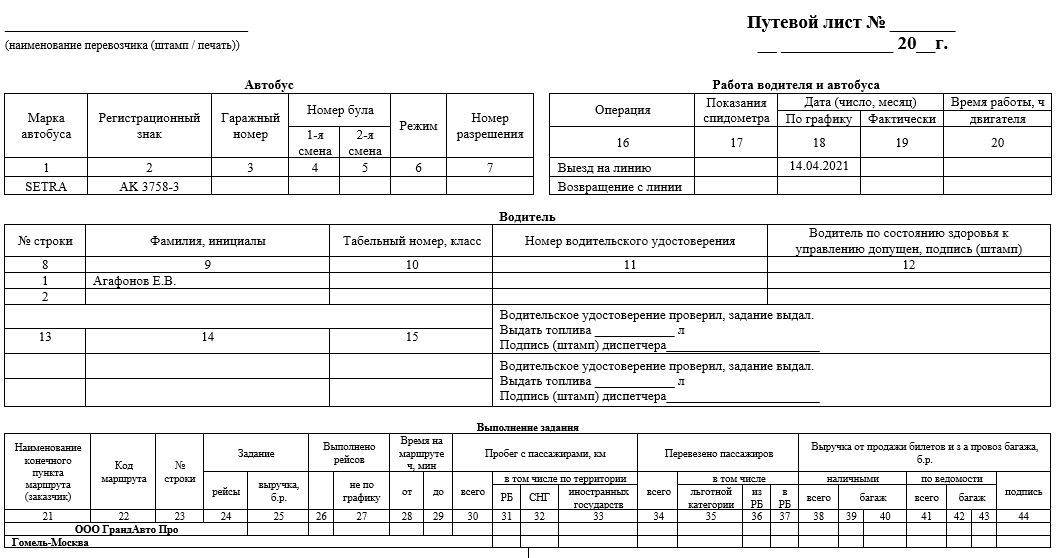


Рисунок 3.10 – Отчет по путевому листу

Из рисунка 3.10 видно, что данные выбранной пользователем записи подставлены в указанный шаблон, в результате чего формируется новый документ.

Для остальных таблиц в базе данных реализованы аналогичные графические элементы и соответствующие методы.

**ЗАКЛЮЧЕНИЕ**

В результате прохождения преддипломной практики была разработана автоматизированная система для формирования отчетов автобусных перевозок, позволяющая пользователю выводить, удалять, редактировать и добавлять данные по водителям, автобусам, маршрутам, путевым листам и расходам для последующего экспорта этих данных в *excel-*документы и *doc*-документы для формирования соответствующих отчетов.

При создании программного обеспечения были углублены знания базовых основ программирования на языке *C#*.

В качестве среды разработки программного комплекса использовалась *Microsoft Visual Studio* с технологией управления версиями *Git.*

Для разработки графического интерфейса использовалась технология *WPF*. Разработанное приложение содержит простой и интуитивно понятный пользовательский интерфейс.

Создание данного приложения способствовало закреплению навыков, полученных во время изучения теоретического материала, и предоставило возможность углубить знания объектно-ориентированного программирования и научиться искать новую информацию.

**Список использованных источников**

1. История *Innowise Group*: *Innowise*. – Электрон. данные. – Режим доступа: https://innowise-group.com/about/. – Дата доступа: 18.04.2021.

2. Охрана труда: Электронные сервисы для системы образования. – Электрон. данные. – Режим доступа: https://sad29orsha.schools.by/pages/ohrana-truda-obespechenija-bezopasnosti-rabotnikov. – Дата доступа: 18.04.2021.

3. Рабочее место при выполнении работ сидя:Фонд актуальных правовых и нормативно-технических документов. – Электрон. данные. – Режим доступа: https://docs.cntd.ru/document/1200003913. – Дата доступа: 18.04.2021.

4. Санитарные правила и нормы: Энергодокумент. – Электрон. данные. – Режим доступа: https://energodoc.by/document/view?id=1467. – Дата доступа: 18.04.2021.

5. Программный комплекс *Bus Commander*: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/bus-commander/. – Дата доступа: 18.04.2021.

6. Программное обеспечение *Transport Manager*: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/tran-sport-manager/. – Дата доступа: 18.04.2021.

7. Облачный сервис *Octobus*: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/octobus/. – Дата доступа: 18.04.2021.

8. Программное обеспечение КВЦ-сервис: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/kvc-servis/. – Дата доступа: 18.04.2021.

9. Мобильное приложение *BusTicketPro*: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/bus-ticketpro/. – Дата доступа: 18.04.2021.

10. Программное обеспечение *Biletion Office*: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/bi-letion-office/. – Дата доступа: 18.04.2021.

11. Авибус: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/avi-bus/. – Дата доступа: 18.04.2021.

12. Программный комплекс Е-Автовокзал: Программы и сервисы для бизнеса. – Электрон. данные. – Режим доступа: https://amastart.com/software/e-atp/. – Дата доступа: 18.04.2021.

13. Пушников, А. Ю. Введение в системы управления базами данных: учебник / А. Ю. Пушников. – Уфа: изд. Башкирского ун-та, 2010. – 108 с.

14. Орлов, С. А. Теория и практика языков программирования: учебник для вузов / С. А. Орлов. – СПб.: Питер, 2013. – 688 с.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Должностная инструкция инженера-программиста**

Общие положения:

* инженер-программист относится к категории специалистов;
* инженер-программист назначается на должность и освобождается от нее приказом генерального директора по представлению технического директора / начальника структурного подразделения;
* инженер-программист подчиняется непосредственно техническому директору или начальнику структурного подразделения;
* на время отсутствия инженера-программиста его права и обязанности переходят к другому должностному лицу, о чем объявляется в приказе по организации;
* на должность инженера-программиста назначается лицо, отвечающее следующим требованиям: высшее профессиональное (техническое) образование, стаж работы от года.

Инженер-программист должен знать:

* руководящие и нормативные материалы, регламентирующие методы разработки алгоритмов и программ и использования вычислительной техники при обработке информации;
* основные принципы структурного программирования;
* виды программного обеспечения;
* технологию автоматической обработки информации и кодирования информации;
* формализованные языки программирования;
* порядок оформления технической документации.

Инженер-программист руководствуется в своей деятельности:

* законодательными актами РБ;
* уставом организации, Правилами внутреннего трудового распорядка, другими нормативными актами компании;
* приказами и распоряжениями руководства;
* настоящей должностной инструкцией.

Инженер-программист выполняет следующие должностные обязанности:

* на основе анализа математических моделей и алгоритмов решения экономических и других задач разрабатывает программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку;
* разрабатывает технологию решения задачи по всем этапам обработки информации;
* осуществляет выбор языка программирования для описания алгоритмов и структур данных;
* определяет информацию, подлежащую обработке средствами вычислительной техники, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и вывода, методы ее контроля;
* выполняет работу по подготовке программ к отладке и проводит отладку;
* осуществляет запуск отлаженных программ и ввод исходных данных, определяемых условиями поставленных задач;
* проводит корректировку разработанной программы на основе анализа выходных данных;
* разрабатывает инструкции по работе с программами, оформляет необходимую техническую документацию;
* определяет возможность использования готовых программных продуктов;
* осуществляет сопровождение внедрения программ и программных средств;
* разрабатывает и внедряет системы автоматической проверки правильности программ, типовые и стандартные программные средства, составляет технологию обработки информации;
* выполняет работу по унификации и типизации вычислительных процессов.

Права инженера-программиста:

* знакомиться с проектами решений руководства предприятия, касающихся его деятельности;
* вносить на рассмотрение руководства предложения по совершенствованию работы, связанной с предусмотренными настоящей инструкцией обязанностями;
* сообщать своему непосредственному руководителю о всех выявленных в процессе осуществления должностных обязанностей недостатках в деятельности предприятия (его структурных подразделениях) и вносить предложения по их устранению.

Ответственность инженера-программиста:

* за невыполнение и/или несвоевременное, халатное выполнение своих должностных обязанностей;
* за несоблюдение действующих инструкций, приказов и распоряжений по сохранению коммерческой тайны и конфиденциальной информации.

На время отсутствия инженера-программиста (отпуск, болезнь, командировка, пр.) его обязанности исполняет лицо, назначенное в установленном порядке, которое несет ответственность за качественное исполнение возложенных на него обязанностей.

Каждому сотруднику организации предоставляется доступ к активам ИТ организации, используемым при реализации бизнес-процессов и процессов внутренней деятельности организации.

Для получения доступа к активам сотрудник должен быть авторизован в ИС организации.

При авторизации сотруднику должен присваиваться уникальный идентификатор (как правило, совпадающий с его фамилией) и предоставляться права доступа к активам.

Права доступа сотрудника к активам должны предоставляться в объеме, необходимом сотруднику для выполнения его функциональных и должностных обязанностей.

Каждому сотруднику организации предоставляются по умолчанию минимальные права доступа к ИТ-активам (ресурсам технологической инфраструктуры и приложениям), которые необходимы ему для коммуникационного взаимодействия с другими сотрудниками, службами и подразделениями организации.

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Листинг программы**

**Код *App.config*:**