Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске
Кафедра систем управления и информационных технологий

Утверждена распоряжением по институту
от 10 октября 2019 № 120-р/с

Допущена к защите
«_____» _____ 2019 г.
Зав. кафедрой СУиИТ
д. т. н., профессор, Першин И. М.
_____

ВЫПУСКНАЯ КВАЛИФИКАЦИОНАЯ РАБОТА
РАЗРАБОТКА МЕТОДОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОРПОРАТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ

Рецензент:
Клименко Андрей Павлович
Руководитель группы отдела технической защиты информации
_____
_

Нормоконтролер:
Чернышев Александр Борисович
Профессор кафедры СУиИТ
_____ _____
(Подпись)

Дата защиты
«___» _____ 2019 г.

Оценка
_____ _____
(Подпись)

Выполнил:
Елисеев Николай Алексеевич
студент 3 курса, **группы** ИНБ-м-оз-171
направления 10.04.01 Информационная безопасность
направленность (профиль) -
Комплексная защита инфокоммуникационных объектов
очно-заочной **формы обучения**

Научный руководитель:
Мартиросян Карина Владиковна
доцент кафедры СУиИТ доцент, к.т.н.

Пятигорск, 2019 г.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АИС – автоматизированная информационная система;

БД – база данных;

ГОСТ – государственный отраслевой стандарт;

ИБ – информационная безопасность;

ИС – информационная система;

ЗК – задача коммивояжёра;

ЛВС – локальная вычислительная сеть;

ООО – Общество с ограниченной ответственностью;

ПК – персональный компьютер;

ПО – программное обеспечение;

СУБД – система управления базами данных;

ЭВМ– электронно-вычислительная машина;

AJAX - Asynchronous Javascript and XML;

DFD – Data Flow Diagrams;

IDEF – ICAM Definition;

IEEE – Institute of Electrical and Electronic Engineers, Международный институт инженеров электротехники и электроники;

ICAM – Integrated Computer Aided Manufacturing;

JSON – JavaScript Object Notation;

HTML – HyperText Markup Language, язык гипертекстовой разметки;

HTTP – HyperText Transfer Protocol, протокол передачи гипертекста;

PDO – PHP Data Object, Объект данных PHP;

RAID – Redundant Array of Independent Disks (избыточных массив независимых дисков);

SaaS – Software as a Service (программное обеспечение как услуга);

XML – Extensible Markup Language.

ВВЕДЕНИЕ

В современном мире сложно представить существование бизнеса без использования сети интернет. Сети интернет используются для обмена файлами, организации частных виртуальных сетей, создания общих ресурсов и т.д. В настоящий момент одним из передовых направлений является создание WEB-приложений.

WEB-приложение – это клиент серверное приложение, в котором взаимодействие клиента с сервером осуществляется при помощи браузера[17]. Данный вид приложений обладает рядом преимуществ: кроссплатформенность, совместимость, более низкие требования к аппаратной части (не требуется выполнять трудоемкие операции на клиенте), простота обновлений для конечных пользователей (обновления происходят на сервер и загружаются у клиента при входе в систему) и т.д. Но в тоже время имеется зависимость от качества интернет соединения и возрастает требования к обеспечению информационной безопасности корпоративных WEB-приложений.

Актуальность темы ВКР подкрепляется тем, что сейчас в организации рабочего процесса предприятий происходит переход к использованию средств, повышающих мобильность пользователей и упрощающих доступ к требуемым информационным ресурсам. К таким средствам относятся WEB-приложения, что ставит задачу обеспечения безопасности WEB-приложений одной из основных задач при их разработке.

В качестве объекта исследования выступает система защиты корпоративного WEB-приложения.

Предметом исследования являются методы обеспечения защиты корпоративных WEB-приложений.

Целью выпускной квалификационной работы «Разработка методов обеспечения информационной безопасности корпоративных веб-приложений» является изучение аспектов повышения уровня безопасности

WEB-приложений, а также их применение на практике, путем разработки модуля защиты в подсистеме по формированию путевых листов.

Задачами ВКР являются:

— анализ угроз корпоративных WEB-приложений;

— анализ методов защиты компонентов ЦУИ-приложений;

— анализ деятельности и структуры организации;

— анализ аппаратного и программного обеспечения информационной системы организации;

— выбор комплекса задач, требующих разработки и внедрения средств защиты;

— обоснование необходимость разработки средств защиты;

— анализ методов решения поставленной задачи;

— модернизация структуры базы данных;

— разработка программного продукта.

В качестве теоретико-методологических основ исследования в данной выпускной квалификационной работе рассматривается ряд литературных источников и научных работ – всего 36 источников. При решении поставленных задач в качестве основополагающих документов используются национальные стандарты Российской Федерации и международные стандарты ISO.

Работа состоит из трех частей: теоретической части, аналитической и проектной.

В первой части выпускной квалификационной работы были даны теоретические основы по теме работы: необходимые определения, классификация информационных систем, уязвимостей и атак на WEB-приложения. Также был произведен анализ средств обеспечения безопасности WEB-приложений.

Во второй части был проведен анализ деятельности организации, выявлена и поставлена задача, сформированы цель и назначение

разрабатываемой подсистемы защиты, осуществлено обоснование выбора проектных решений.

В проектной части была описана структура модернизированной базы данных, приведены характеристики результирующей информации, описаны модули подсистемы после модернизации и их взаимосвязь, описан интерфейс подсистемы.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Основные понятия

1.1.1 Понятие корпоративной информационной системы

Информационная система – это система, которая предназначена для обработки, хранения и представления информации в необходимой форме для решения поставленных задач. Информационная система включает в себя средства и методы для работы с информацией, а также персонал[8].

В современном понимание информационная система подразумевает использование вычислительной техники в качестве основного средства обработки информации, то есть программные, программно-аппаратные и аппаратные средства.

Информационные системы могут решать широкий спектр задач разного рода, поэтому необходима классифицировать их. Классификация информационных систем показана на рисунке 1.1.

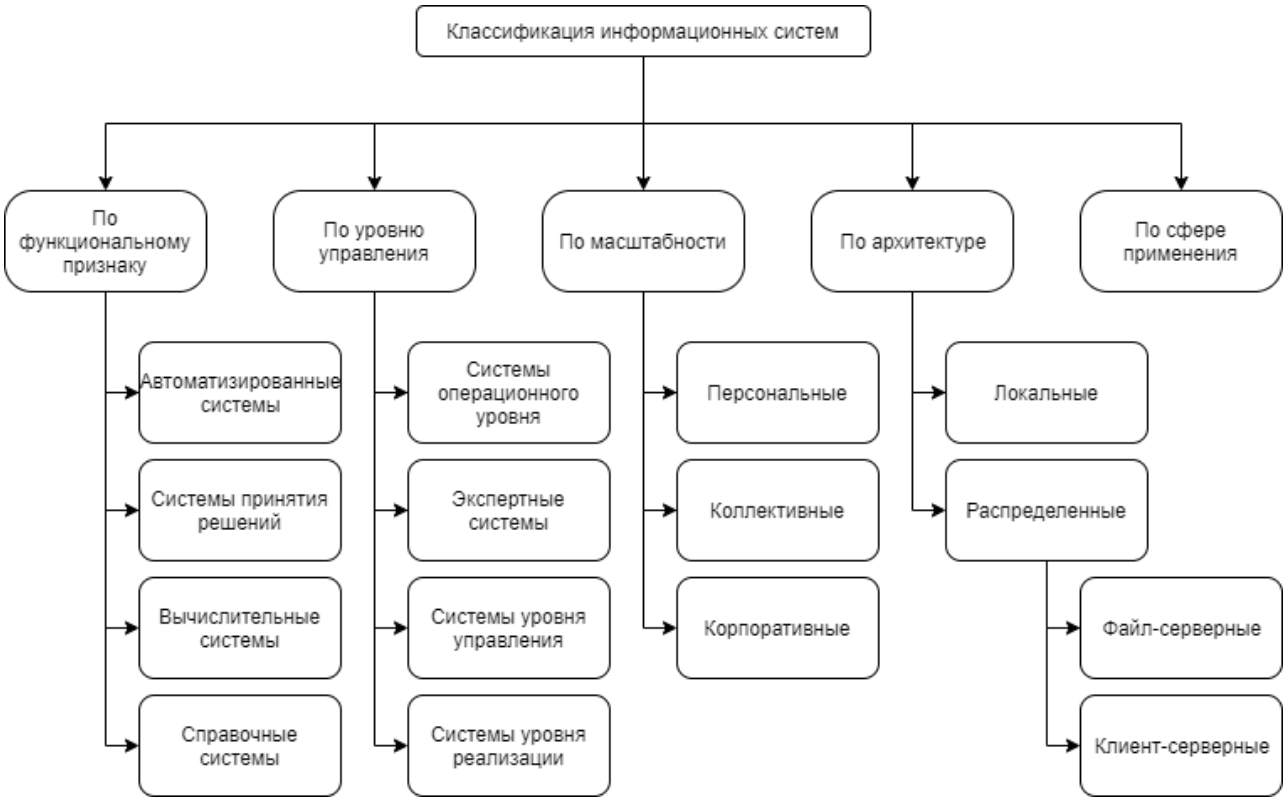


Рисунок 1.1 – Классификация информационных систем

Существует несколько видов классификации информационных систем:

- По функциональному признаку:
  - автоматизированные системы;
  - системы принятия решений;
  - вычислительные системы;
  - справочные системы, системы обучения);
- по уровню управления:
  - системы операционного уровня – бухгалтерские, банковские, системы бронирования, обработки заказов и другие;
  - экспертные системы;
  - системы уровня управления (тактические);
  - системы уровня реализации (стратегические);
- по сфере применения (медицинские, геоинформационные и т.п.);
- по архитектуре:
  - локальные (настольные);
  - распределенные (файл-серверные, клиент-серверные)
- по масштабности:
  - персональные (индивидуальные);
  - коллективные;
  - корпоративные.

Также информационные системы могут совмещать в себе несколько типов из данной классификации, в таком случае принято говорить об разделении информационной системы на подсистемы.

Информационная подсистема – часть информационной системы, реализующая некоторую часть функционала системы и связанная с другими подсистемами.

В рамках данной выпускной квалификационной работы рассматриваются корпоративные информационные системы, то есть системы, предназначенные для автоматизации хозяйственной деятельности предприятия.

1.1.2 Корпоративные WEB-приложения

Развитие глобальной сети интернет и увеличение доступности ресурсов в ней, привело к расширению спектра задач, решаемых с использованием WEB-технологий. В связи с этим получил распространение особый вид приложений — WEB-приложения, от которых напрямую зависит функционирование бизнес-процессов многих организаций[16].

WEB-приложение — это клиент-серверное приложение, где в качестве клиента выступает браузер, который отображает пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. А серверная часть представляет собой WEB-сервер, обрабатывающий запросы клиентов. Взаимодействие между клиентом и сервером, как правило, осуществляется посредством протокола HTTP[19]. Архитектура WEB-приложений имеет три уровня[4], которые показаны на рисунке 1.2.

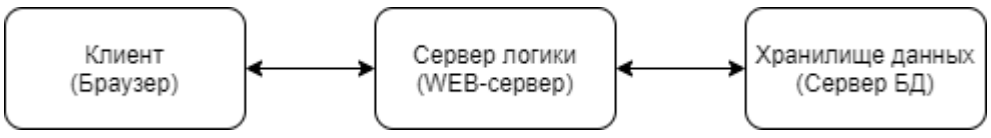


Рисунок 1.2 — Архитектура WEB-приложения

Основными особенностями, повлиявшими на распространение WEB-приложений, являются:

1. Доступность — данный вид приложений не привязан к определенному терминалу или локальной сети, доступ может осуществляться из любой точки Земного шара, где присутствует соединение с Интернет.

2. Кроссплатформенность — клиенту достаточно иметь браузер, соответствующий стандартам, а операционная система и тип устройства не имеют никакого значения.

3. Не требует установки и настройки отдельного приложения на клиентском устройстве.

4. Автоматическое обновление — клиент всегда работает с самой актуальной версией приложения, т. к. все обновления происходят на стороне сервера.

1.2 Исследование методов обеспечения информационной безопасности корпоративных WEB-приложений

1.2.1 Угрозы безопасности корпоративных WEB-приложения

В процессе эволюции WEB-приложений разработчики сталкивались с рядом проблем, среди которых особняком стоит проблема безопасности WEB-приложений. Проблемы безопасности вытекают из основных особенностей WEB-приложений, например, из-за невозможности изолировать WEB-приложение от попыток несанкционированного доступа извне ввиду их доступности.

Актуальность проблем безопасности WEB-приложений подкрепляется тем, что в них используется конфиденциальная информация, а также осуществляются бизнес-процессы компании, например:

— хранение и обработка персональные данных сотрудников и клиентов;

— сведения о финансовых операциях компании;

— сведения, составляющие коммерческую тайну;

— сведения о взаимодействии, а также взаимодействие с клиентами компании;

— взаимодействие между сотрудниками и отделами;

— другие сведения и процессы, зависящие от рода деятельности компании[34].

Согласно классификации международной организации Web Application Security Consortium (WASC) выделяется 6 классов атак на WEB-приложения:

1. Аутентификация – класс атак, который содержит атаки, направленные на методы проверки подлинности пользователя.
2. Авторизация – класс атак, направленных на методы проверки легитимности действий пользователя.
3. Атаки на клиентов – атаки, которые подразумевают подмену легитимного содержимого, при взаимодействии пользователя с приложением.
4. Выполнение кода – реализуется злоумышленниками путем внедрения вредоносного кода на стороне WEB-сервера.
5. Разглашение информации – атаки, направленные на выяснение информации об инфраструктуре WEB-сервера, чтобы в дальнейшем иметь представление о возможных уязвимостях и планированию реализации атак.
6. Логические атаки – эксплуатация функциональных возможностей приложения при выполнении определенных действий.

Среди разработчиков появились рекомендации по обеспечению безопасности WEB-приложений, которые вылились в проект под названием: Open Web Application Security Project (OWASP).

OWASP – это открытый проект обеспечения безопасности WEB-приложений, который включает в себя корпорации, образовательные организации и индивидуальных разработчиков, которые совместными усилиями формируют статьи, рекомендации и учебные пособия, находящиеся в свободном доступе и рекомендуемы при разработке WEB-приложений. Также проект включает в себя ряд тренировочных задач и инструменты для анализа безопасности WEB-приложений.

OWASP на протяжении всего своего существования занимается изучением наиболее опасных уязвимостей и ежегодно публикует отчет.

Рассмотрим наиболее опасные угрозы безопасности WEB-приложений, согласно исследованиям OWASP Top 10 - 2017 [34]:

1. Инъекции (Injection) — внедрение в запросы к базе данных кода, дополняющего данный запрос и дающего злоумышленнику неавторизованный доступ к базе данных.

2. Уязвимости аутентификации (Broken Authentification) – распространенная уязвимость, связанная с недостаточно проработанной системой валидации пользователей в приложении, приводит к получению неавторизованного доступа.

3. Незащищенность важных данных (Sensitive Data Exposure) — многие приложения не используют механизмов для защиты передаваемых данных, таких как, например, HTTPS.

4. Внедрение внешних сущностей в XML (XML External Entities) — вид инъекции, основанный на внедрении в XML-запрос к серверу атрибутов и сущностей, позволяющих получить неавторизованный доступ к данным.

5. Небезопасный контроль доступа (Broken Access Control) — уязвимость в методах авторизации, позволяющие злоумышленнику получить повышенные привилегии.

6. Небезопасная конфигурация (Security Misconfiguration) — WEB-приложение — это сложная система, состоящая из многих компонентов, таких как WEB-сервер, СУБД и др. Неверная конфигурация одного из компонентов может привести к серьезным проблемам с безопасностью всего приложений.

7. XSS(Cross-Site Scripting) – внедрение (инъекция) вредоносного кода в HTTP-ответ, получаемый клиентом и выполняющийся на стороне клиента.

8. Небезопасная десериализация (Insecure Deserialization) — десериализация преобразует последовательность бит в структурированные данные, зачастую на данном этапе не уделяется достаточно внимания безопасности, например, отсутствует валидация типов данных, что приводит к их подмене.

9. Использование компонентов с известными уязвимостями (Using Components with Known Vulnerabilities) — зачастую при разработке WEB-приложений используются библиотеки, фреймворки и компоненты сторонних разработчиков, которые могут содержать различные недостатки (уязвимости), в связи с этим важно использовать самые актуальные версии, в которых исправляются известные уязвимости.

10. Недостаточное журналирование и мониторинг (Insufficient Logging&Monitoring) — для своевременного обнаружения несанкционированного доступа, утечки информации и т. д., необходимо использовать средства автоматизированного мониторинга трафика, а также журналирования, которые помогут понять сущность атаки и разработать в кратчайшие сроки устранить уязвимости, а также вернуть работоспособность и состояние WEB-приложений.

Данный список уточняется OWASP в соответствии с развитием модели поведения нарушителя и актуальности той или иной проблемы безопасности WEB-приложений. Но, к сожалению, некоторые уязвимости (например, SQL-инъекции) находятся на вершине списка долгое время, хотя уже давно разработаны методы, закрывающие данную уязвимость. Это связано с тем, что не всегда разработчики уделяют достаточно внимания безопасности WEB-приложений.

В рамках проекта OWASP были разработаны рекомендации, призванные повысить безопасность WEB-приложений:

— необходимо ознакомиться и следить за обновлениями документов OWASP;

— использовать методологии тестирования WEB-приложений, направленные на поиск уязвимостей (например, OWASP Testing Project);

— регулярно обновлять программное обеспечение WEB-сервера;

— следить за корректной настройкой сетевых устройств, служб и программного обеспечения;

— следить за обновлениями используемых в приложении фреймворков и библиотек, и своевременно устранять найденные в них уязвимости;

— использование протоколов с шифрованием (HTTPS);

— повсеместное использование средств обнаружения атак, мониторинга активности, журналирования и системы транзакций.

Данные рекомендации позволят значительно повысить уровень безопасности WEB-приложений, что приведет к снижению рисков компаний, использующих WEB-приложения.

1.2.2 Методы защиты баз данных корпоративных WEB-приложений

База данных приложения содержит совокупность данных, с которыми пользователь взаимодействует в процессе работы с информационной системой[8]. Практически всегда эти данные представляют ценность для организации, что повышает риск взлома и утраты этих сведений, поэтому защита баз данных в корпоративных приложениях выходит на первый план.

Основные методы защиты баз данных[28]:

1. Защита паролем – самый простой, но в тоже время достаточно эффективный способ защиты баз данных, который позволяет достичь приемлемого уровня информационной безопасности.

2. Криптографические методы защиты (шифрование) – использование криптографических методов сокрытия информации позволит предотвратить угрозы, связанные с кражей носителей базы данных и перехвата данных при передаче по коммуникационным каналам.

3. Использование инструментов для распределения прав доступа к объектам базы данных – гибкая система, которая позволяет системному администратору разграничивать доступ пользователей (или групп пользователей) к различным объектам (таблицам, записям, представлениям и т.д.).

4. Контроль действий пользователей баз данных – позволяет регистрировать операции, совершаемые пользователями баз данных, что позволяет значительно повысить уровень безопасности баз данных, за счет своевременного реагирования на инциденты и их расследования. Также в некоторых случаях возможна реализация отката транзакций, чтобы вернуть систему к исходному состоянию.

5. Резервное копирование – позволяет восстановить базу данных в случае аппаратного или программного сбоя, которые могут привести к полному или частичному уничтожению информации в базе данных. Резервное копирование производится с определенной периодичностью, зависящей от характеристик и частоты обновления информации в ней. В некоторых случаев необходимо выполнение резервного копирования в реальном времени, когда потеря даже минимального количества информации может быть критична.

1.2.3 Методы защиты сетевой инфраструктуры корпоративных WEB-приложений

Под сетевой инфраструктурой понимается совокупность оборудования и программного обеспечения, которое применяется для взаимодействия между компонентами информационной системы в рамках обмена данными в процессе выполнения задач, для которых используется система[21].

Сетевая инфраструктура состоит из пассивного оборудования (патч-корды, патч-панели, шкафы, среды передачи данных и т.д.), активного оборудования (коммутаторов, маршрутизаторов, медиаконвертеров и т.д.), программного обеспечения, а также периферийного оборудования и средств вычислительной техники.

Для защиты сетевой инфраструктуры применяются[32]:

— Технические средства защиты (инженерно-технические решения, позволяющие ограничить доступ к инфраструктуре и создающие дополнительные рубежи на пути действия злоумышленников);

— Криптографические методы защиты (шифрование, функции хеширования);

— Методы разграничения доступа (как на физическом, так и на программном уровне);

— Протоколирование и аудит;

— Защита от вирусов;

— Экранирование (использование сетевых экранов);

— Изоляция критичных сегментов сети.

1.3 Средства обеспечения информационной безопасности WEB-приложений

Задача по обеспечение информационной безопасности WEB-приложений подразумевает решение ряда проблем, связанных со спецификой функционирования такого вида приложений. Так как само по себе WEB-приложение подразумевает использование сети интернет и доступность из любой точки мира (кроме случаев, в которых по каким-либо причинам установлены территориальные ограничения), то нельзя просто изолировать приложение.

В связи с этим применение обычных межсетевых экранов помогает защититься только от видов атак, направленных на различные сервисы в сети предприятия, в то время как трафик по HTTP и HTTPS (80 и 443 порты, соответственно) будет пропущен в соответствии с правилами, заданными в параметрах файрволла[23]. Поэтому для WEB-приложений на рынке представлены комплексы, который получили название Web Application Firewall (WAF).

Web Application Firewall – это узкоспециализированный тип программного файрволла, который разработан с учетом специфики работы WEB-приложений. Он подразумевает анализ трафика на предмет выявления

подозрительной активности и принятие мер реагирования на подозрительный траффик[14].

WAF основан на применении сигнатур атак, автоматическом обучении, списках подозрительных IP-адресов и т.д.

Так как средство специализируется на применении с WEB-приложениями, то оно анализирует как траффик по протоколам HTTP/HTTPS, так и траффик, организованный на применении способов обмена данными поверх HTTP/HTTPS, таких как XML, JSON и другие. Что выгодно выделяет его на фоне обычных систем обнаружения и предотвращения вторжений (IDS/IPS).

В настоящее время существует большое разнообразие WAF систем, в том числе и Open Source решение, развиваемое силами сообщества проекта OWASP – ModSecurity[34]. Некоторые коммерческие решения:

— Barracuda Networks WAF;

— Citrix Netscaler Application Firewall;

— F5 Big-IP ASM;

— Qualys WAF;

— Sophos XG Firewall;

— PT AF (сертифицирован ФСТЭК);

— Imperva SecureSphere Web Application Firewall (сертифицирован ФСТЭК).

Также широкое распространение получают WAF, основанные на применении облачных технологий, такие как:

— Amazon Web Services AWS WAF;

— Alibaba Cloud;

— Microsoft Azure Application Gateway with WAF;

— IBM Cloud Internet Services WAF.

Также OWASP рекомендует использовать средства анализа уязвимостей, которые позволяют на еще этапе разработки выявить проблемы в подсистеме безопасности приложения. К таким средствам относятся[27]:

— SQLmap (сканер, который автоматизирует процесс тестирования приложение на предмет возможности реализации SQL-инъекций, сканер поддерживает большинство СУБД);

— Metasploit (позволяет обнаружить целый ряд уязвимостей в WEB-приложении: SQL-инъекции, XSS и другие);

— WebCruiser Web Vulnerability Scanner (сканер широкого спектра уязвимостей с графическим интерфейсом и встроенным браузером);

— NetSparker (сканер с поддержкой Javascript и современных реализаций стандартов HTML).

1.4 Технологии разработки защищенных корпоративных WEB-приложений

Технология разработки защищенных WEB-приложений подразумевает должное внимание к вопросам безопасности приложения на всех этапах жизненного цикла: проектировании, разработке, тестировании, эксплуатации, поддержке[25].

При разработке WEB-приложений особое внимание уделяется данным, поступающим в приложение от пользователя, необходимо полагать что любые данные могут быть скомпрометированы или изначально представлять опасность для приложения. Поэтому технологии разработки включают ряд важных этапов и процедур, позволяющих снизить вероятность реализации угрозы.

На этапе проектирования приложения необходимо определить возможные угрозы, способы их реализации и отразить их в документации. Процесс моделирования угроз заключается в анализе входных данных, бизнес-процессов и создании модели нарушителя.

Во время разработки приложения необходимо уделить внимание используемым программным и аппаратным средствам: оборудовании, операционной системе, программном обеспечении сервера и используемым компонентам (фреймворки и библиотеки).

Процесс тестирования приложения идет является важной частью разработки и ввода в эксплуатацию. Тестирование подразумевает использование специальных средств (анализаторов кода, сканеров уязвимости и т.д.), а также создание тестов, реализующих угрозы, выявленные на этапе проектирования.

Существует несколько два основных подхода к тестированию WEB-приложений[15]:

— Тестирование «Черного ящика» - подразумевает что атакующий не знает об структуре приложения и не имеет доступа к исходным кодам. Все входные воздействия происходят с использованием пользовательских методов и функциональных возможностей приложения;

— Тестирование «Белого ящика» - подразумевается, что тестирование проводится со знанием внутренней структуры приложения. Тестирование идет путем анализа исходных кодов приложения.

После ввода в эксплуатацию наступает фаза поддержки программного обеспечения. Разработчики и администраторы системы должны поддерживать приложение в актуальном состоянии. Своевременно обновлять компоненты приложения (особенно в случае выявления потенциальных уязвимостей в старых версиях). Реагировать на инциденты, связанные с нарушением безопасности приложений: проводить анализ каждого инцидента и вырабатывать стратегию по ликвидации последствий и предотвращению рецидивов.

Выводы по разделу.

В данной главе был изучен теоретический материал по теме выпускной квалификационной работе. Рассмотрены типы информационных систем,

принципы работы WEB-приложений, а также возможные уязвимости WEB-приложений. Были рассмотрены рекомендации, методы и средства, призванные повысить уровень защищенности корпоративных WEB-приложений.

2 АНАЛИТИЧЕСКАЯ ЧАСТЬ

2.1 Организация информационной безопасности корпоративных веб-приложений на примере подсистемы формирования путевых листов

2.1.1 Характеристика предприятия и его деятельности

Общество с ограниченной ответственностью «Каскад» было зарегистрировано в 2011 году и более 7 лет предлагает услуги в области защиты информации. Компании зарегистрирована по адресу г. Пятигорск, ул. Московская, д. 68А. Основным видом деятельности компании является Производство электромонтажных работ. Также ООО «Каскад», работает еще по 23 направлениям, среди них следует выделить:

— Разработка компьютерного программного обеспечения;

— Технические испытания, исследования, анализ и сертификация;

— Деятельность по техническому контролю, испытаниям и анализу прочая;

— Деятельность в области защиты информации;

— Деятельность систем обеспечения безопасности;

— Ремонт компьютеров и периферийного компьютерного оборудования;

— Деятельность по обработке данных, предоставление услуг по размещению информации и связанная с этим деятельность;

— Деятельность, связанная с использованием вычислительной техники и информационных технологий, прочая.

ООО «Каскад» сотрудничает с крупнейшими вендорами России:

— Dr. Web;

— Kaspersky Lab;

— Код безопасности;

— Эшелон;

— Инфортекс;

— Аладдин;

— Конфидент;

— КриптоПро.

Так как организация занимается деятельностью по защите информации на объектах информатизации, а также защитой государственной тайны (по уровню доступа до грифа «Совершенно секретно», включительно), которые требуют лицензирования, то у неё имеется 6 лицензий:

— №ЛСЗ0014838 16024В от 22 июня 2017 г. На деятельность по выявлению электронных устройств, предназначенных для негласного получения информации (за исключением случая, если указанная деятельность осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя) (бессрочная, выдана ФСБ РФ);

— №26.01.05.002.Л.000004.03.17 от 28 марта 2017 г. На деятельность в области использования источников ионизирующего излучения (генерирующих) (за исключением случая, если эти источники используются в медицинской деятельности);

— № ЛСЗ0002944 116Н от 24 октября 2016 г. На разработку, производство, распространение шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнение работ, оказание услуг в области шифрования информации, техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется для

обеспечения собственных нужд юридического лица или индивидуального предпринимателя);

— №ГТ 0077086 950 от 24 марта 2016 г. На деятельность, связанную с защитой государственной тайны (ФСБ России, ФСТЭК России, СВР России, Минобороны России);

— №2726 от 12 октября 2015 г. На деятельность по технической защите конфиденциальной информации;

— №1465 от 12 октября 2015 г. На разработку и производство средств защиты конфиденциальной информации.

Во главе предприятия стоит директор, он несет ответственность за деятельность предприятия и организует работу всех отделов предприятия.

Часть отделов предприятия относятся к режимно-секретным подразделениям.

Среди отделов предприятия следует выделить отдел защиты информации, который включает три подразделения:

— Отдел технической защиты информации, который занимается проектированием и внедрением некриптографических (инженерно-технических и аппаратных) средств защиты информации;

— Отдел программных средств защиты, занимается разработкой программных средств защиты, связанных с использованием криптографических методов защиты информации, а также проектирование и внедрение существующих программных и программно-аппаратных комплексов;

— Отдел по работе с государственной тайной – решает задачи, связанные с организацией, ведением и защите документооборота при работе с информацией, которая составляет государственную тайну.

Сотрудники отдела аттестации технических средств занимаются аттестацией технических средств защиты информации, электронной вычислительной техники и прочих технических средств, связанных с

обработкой информации. Аттестация происходит по методикам ФСТЕК и ФСБ России и необходима для осуществления деятельности, которая связана с государственной тайной, а также при работе с конфиденциальной информацией, например, с персональными данными. По итогам процедуры аттестации сотрудниками ООО «Каскад» выдается Аттестат соответствия по требованиям безопасности информации, либо предписание на устранение недостатков и направление на повторную аттестацию после устранения недочетов.

Остальные отделы играют важную роль в хозяйственной деятельности предприятия. Организационная структура со всеми отделами представлена на рисунке 2.1.

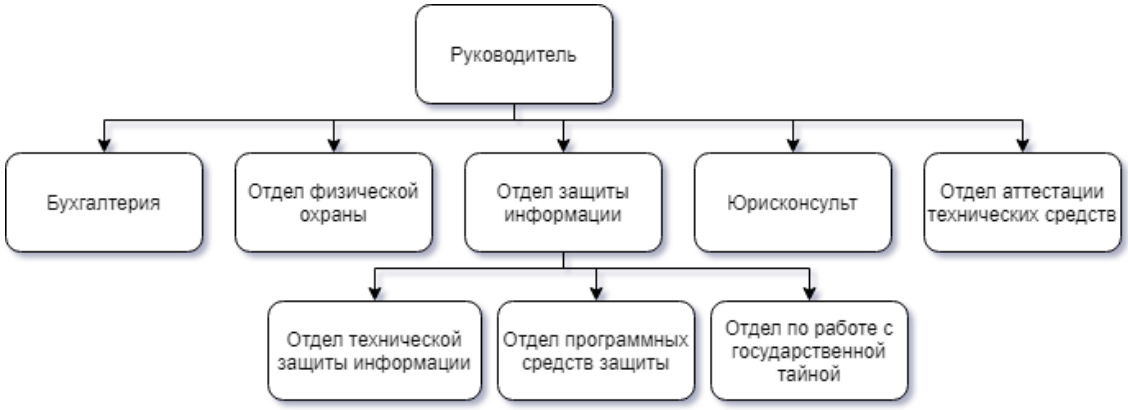


Рисунок 2.1 – Организационная структура ООО «Каскад», г. Пятигорск

Офис компании в городе Пятигорске по адресу регистрации. В составе ООО «Каскад» имеется 16 помещений, среди которых есть помещения отдела технической защиты информации, в которых находятся автоматизированные рабочие места, аттестованные по требованиям информационной безопасности.

Также особым образом выделяются помещения для проведения аттестаций рабочих мест и оборудования, а также помещения для работы с государственной тайной с двумя специализированными рабочими местами.

В составе предприятия насчитывается 21 рабочее место и сервера, которые объединены в локальную сеть. Для обеспечения соответствующего

уровня безопасности используются системы разграничения доступа, межсетевые экраны и организованы демилитаризованные зоны, там, где это необходимо. Рабочие места, предназначенные для работы с государственной тайной выделены в отдельную сеть, с организацией специализированного защищённого канала связи.

2.1.2 Программная и техническая архитектура ИС предприятия

Для функционирования системы спутникового мониторинга транспорта на предприятии располагается два телематических сервера, предназначенный для сбора, хранения и обработки данных(телеметрии), поступающей от объектов мониторинга.

Так как объем телеметрии от объектов мониторинга достаточно большой и процесс обработки информации часто связан со сложными вычислениями, то сервера обладают достаточной производительность для работы с ней. Подробные технические характеристики телематического сервера №1 и №2 представлены в таблице 2.1 и 2.2.

Таблица 2.1 -Технические характеристики телематического сервера №1

		Кол-во	Производитель	Модель
Сервер №1	ЦПУ	1	Intel	Xeon E3-1230
	ОЗУ	4	Kingston	KVR1600D3D4R11S/8G – 8ГБ
	ГПУ	1	Intel	Integrated
	Жесткий диск	2	Western Digital	WD RE WD1003FBYZ – 1ТБ

Таблица 2.2 -Технические характеристики телематического сервера №2

		Кол-во	Производитель	Модель
Сервер №2	ЦПУ	1	Intel	Xeon E3-1230
	ОЗУ	4	Kingston	KVR1600D3D4R11S/8G – 8ГБ
	ГПУ	1	Intel	HD4000
	Жесткий диск	2	Western Digital	WD RE WD20EFRX – 2ТБ

Для повышения отказоустойчивости диски на всех серверах объединены в RAID 1, который представляет собой зеркальный дисковый массив, что позволяет сохранить данные, даже в случае выхода из строя одного из дисков массива.

Помимо телематического сервера на предприятии располагается файловый сервер, используемый для организации общего доступа к файлам и прочим сетевым ресурсам. На этом сервере установлено программное обеспечение 1С:Бухгалтерия 8.3. Технические характеристики сервера представлены в таблице 2.3.

Таблица 2.3 -Технические характеристики файлового сервера

	Кол-во	Производитель	Модель
Центральный процессор	1	Intel	Core i5 2500
ОЗУ	2	Kingston	KVR1600D3D4R11S/8G – 8ГБ
Графический процессор	1	Intel	HD4000
Жесткий диск	1	Western Digital	WD Caviar Blue WD10EALS – 1ТБ

Для взаимодействия между отделами, а также для доступа к серверам на предприятии организована локальная вычислительная сеть (ЛВС), охватывающая все предприятие. Схема ЛВС показана на рисунке 2.2.

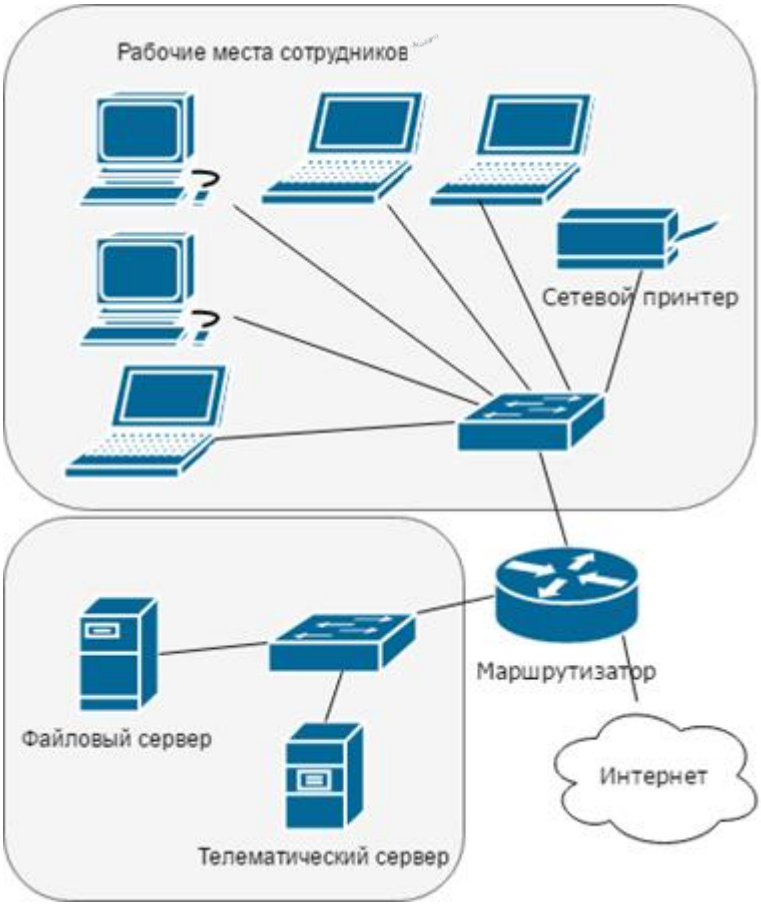


Рисунок 2.2 – Локальная вычислительная сеть

Из схемы следует, что между провайдером и устройствами ЛВС располагается маршрутизатор. Доступ в интернет осуществляется по протоколу PPPoE (Point-to-point protocol over Ethernet). Маршрутизатор также выполняет функции сетевого экрана.

Для изолирования сервера и повышения безопасности ЛВС [22] предприятия разделено на две подсети:

— первый сегмент сети включает все компьютеры отделов предприятия, а также файловый сервер;

— во втором сегменте сети располагаются телематические сервера, в дальнейшем возможно увеличение количества телематических серверов с целью балансировки нагрузки.

Телематический сервер работает под управление операционной системы GNU/Linux. В качестве дистрибутива GNU/Linux был выбран Debian Server 8. Основанием для выбора данного дистрибутива послужила рекомендация разработчиков программного обеспечения, используемого предприятием для организации мониторинга транспорта.

Система спутникового мониторинга базируется на программном обеспечении Wialon, которое установлено на телематических серверах.

Wialon – многофункциональная система GPS/GLONASS мониторинга транспорта, мобильных и стационарных объектов[]. Разработчиком системы является белорусская компания Gurtam, базирующаяся в Минске. Первая версия системы появилась в 2004 году. В настоящее время существует несколько вариантов системы:

— версия для операторов систем спутникового мониторинга (Wialon Pro, Wialon Local);

— SaaS, базирующаяся на серверах разработчиков системы (Wialon Hosting).

Система Wialon получила широкое распространения, благодаря своей политике внедрения поддержки новых устройств, а также постоянному развитию и расширению функционала в соответствии с потребностями рынка. Система используется в более чем 100 странах для мониторинга более 1 000 000 объектов[].

Предприятие использует две версии системы спутникового мониторинга: Wialon PRO 1401 на первом сервере и Wialon Local 1804 на втором сервере.

Wialon Pro представляет собой WEB-приложение, которое располагается на телематическом сервере. В качестве WEB-сервера используется nginx. Для хранения данных используется

высокопроизводительная встраиваемая нереляционная СУБД Oracle Berkeley DB, она отличается возможностью работы с множеством потоков и с базами данных размером до 256Тб.

Wialon Local также является WEB-приложением, которое располагается на телематическом сервере, но является более новой разработкой Gurtam с обновленным интерфейсом и расширенным функционалом. Также система базируется на Node.js, что положительно влияет на скорость работы. В качестве WEB-сервера используется nginx. Для хранения данных используется высокопроизводительная встраиваемая нереляционная СУБД, Разработанная Gurtam.

Система GLONASS/GPS мониторинга Wialon Pro обладает следующими особенностями[13]:

— Отсутствие необходимости установки клиентского приложенияна конечном устройстве, за счет применения web-интерфейс;

— поддержка различных языков интерфейса;

— на выбор пользователя предоставляется более 10 источников карт, в том карты Google Maps, Яндекс.Карты и карты от разработчиков системы - Gurtam Maps;

— поддержка более 1000 типов устройств от более, чем 400 производителей;

— присутствует биллинговая система, которая разграничивать доступ пользователям системы в зависимости от тарифного плана и производить автоматическую блокировку, в случае исчерпания средств на счете клиента;

— возможность интеграции с другими системами и приложениями, посредством технологии ActiveX или с использованием модуля Wialon SDK.

Также система обладает широкими функциональными возможностями, в том числе:

— создание различных датчиков для объекта мониторинга (датчик уровня топлива, датчик мгновенного расхода топлива, датчик зажигания и другие);

— построение трека (передвижения) объекта за указанный период времени, с возможностью ускоренного проигрывания трека;

— создание отчетов, с возможностью конфигурирования под индивидуальные нужды отдельного клиента;

— настройка уведомлений о поведении объекта (превышение скорости, изменение показателей датчиков и т.п.), с возможностью отправки уведомления по e-mail или sms, а также отображения всплывающего окна в web-интерфейсе системы;

— возможность ретранслирования (пересылки) данных, поступающих от объекта мониторинга на другой сервер или в другую систему по различным протоколам.

2.1.3 Характеристика предметной области

Большинство пользователей системы мониторинга – это разного рода автотранспортные предприятия (АТП). Этот факт обусловлен тем, что для АТП особым образом стоит учет работы автотранспорта. Применение систем спутникового мониторинга транспорта позволило упростить ведение этого учета, а также осуществлять контроль за подвижными составами, что повлияло на эффективность деятельности предприятий в лучшую сторону.

Системы спутникового мониторинга основаны на использовании глобальных навигационных спутниковых систем (GPS, GLONASS, Beidou, Galileo) при помощи которых навигационное оборудование (терминал, трекер), установленное на устройстве определяет текущее местоположение и посредством сотовых сетей GSM отправляет данные о местоположении и прочие данные о состоянии устройства на телематический сервер, на котором установлено специализированное программное обеспечение (например,

Wialon). Пользователь системы спутникового мониторинга может получить данные с телематического сервера посредством сети интернет, или обратившись напрямую к базе данных системы (способ получения данных может отличаться в зависимости от архитектуры системы). Схема работы системы спутникового мониторинга показана на рисунке 2.3.

Рисунок 2.3 – Схема работы систем спутникового мониторинга

Одним из самых трудозатратных процессов на АТП - процесс составления маршрута. АТП, которые ежедневно осуществляют развоз продукции по множеству пунктов, пользуются возможностями подсистемы по автоматизации процесса составления маршрута и выдачи путевого листа автомобиля.

В связи с этим повышение уровня безопасности подсистемы является приоритетной задачей.

2.1.4 Определение информации, подлежащей защите

При создании системы защиты одним из основных этапов разработки является определения перечня информации, которую необходимо защищать. Для того чтобы понять какая информация вводится в систему нужно необходимо определить документы, на основе который пользователи вводят

исходные данные. Основными пользователями системы являются частные автотранспортные предприятия.

Автотранспортное предприятие (АТП) – это организация, осуществляющая перевозки автомобильным транспортом, а также хранение, техническое обслуживание и ремонт подвижного состава[6].

Первичным документом учета работы подвижного состава на АТП является путевой лист. На основе путевого листа осуществляется списание горюче-смазочных материалов (ГСМ), рассчитывается заработная плата водителей. Поэтому для АТП особым образом стоит достоверность заполнения путевых листов.

Существует различные виды путевых листов: путевой лист грузового автомобиля, путевой лист автобуса, путевой лист легкового автомобиля, путевой лист легкового такси и другие.

Подсистема ориентирована на организации, которые специализируются на развозе грузов, поэтому подсистема формирует путевой лист грузового автомобиля.

Существует несколько форм путевых листов грузового автомобиля[1]:

— форма 4-П, которая применяется при условии оплаты работы автомобиля по повременному тарифу и рассчитана на одновременное выполнение перевозок грузов до двух заказчиков в течение одного рабочего дня (смены) водителя;

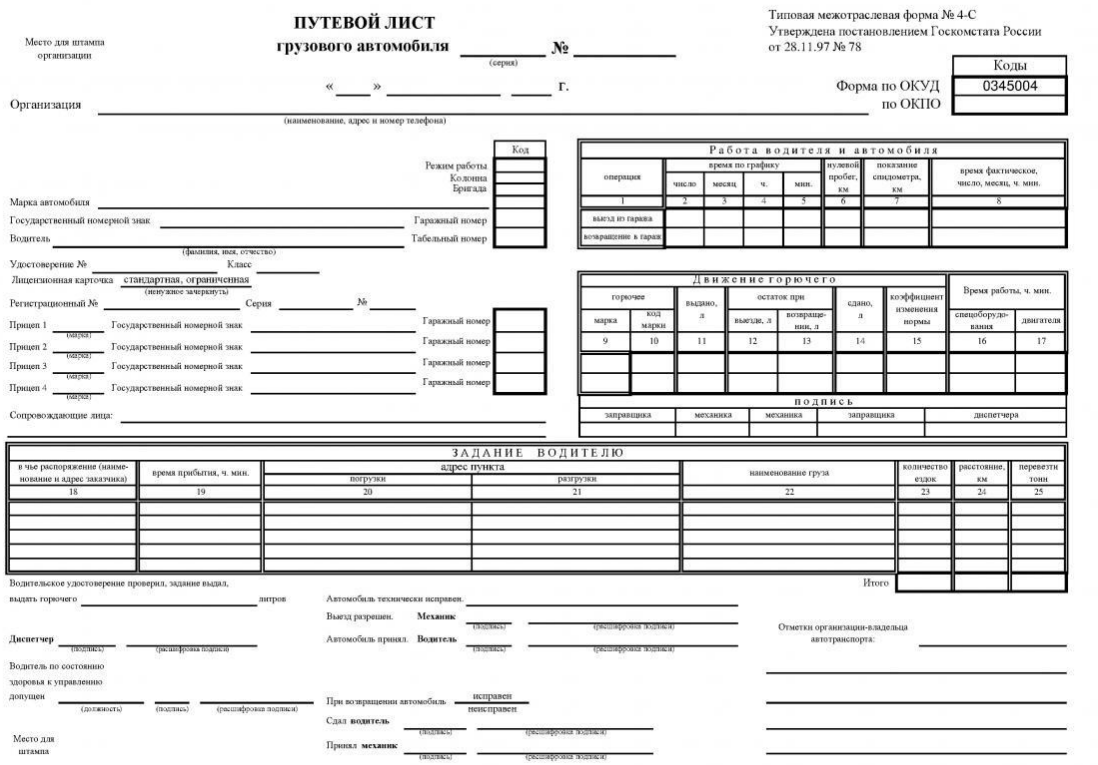
— форма 4-С, применяется при осуществлении перевозок грузов при условии оплаты работы автомобиля по сдельным расценкам;

— форма 4-М, применяющаяся при осуществлении международных перевозок.

Так как перевозки осуществляются в пределах Российской Федерации, то форма 4-М не рассматривается в данной задаче.

Форма 4-П связана с многократным движением автомобиля от пункта загрузки к заказчику, соответственно для данного случая задача автоматизации составления маршрута не имеет смысла.

В соответствии с этим в подсистеме формируется путевой лист по форме 4-С. Образец формы 4-С представлен на рисунке 2.4.



ПУТЕВОЙ ЛИСТ
грузового автомобиля _____ № _____
(серия)

Типовая межотраслевая форма № 4-С
Утверждена постановлением Госкомстата России
от 28.11.97 № 78

Место для штампа организации

«____» _____ ____ г.

Коды
Форма по ОКУД 0345004
по ОКПО

Организация _____
(наименование, адрес и номер телефона)

Код
Режим работы
Колонна
Бригада

Марка автомобиля _____
Государственный номерной знак _____ Гаражный номер
Водитель _____ Табельный номер
(фамилия, имя, отчество)
Удостоверение № _____ Класс _____
Лицензионная карточка стандартная, ограниченная
(ненужное зачеркнуть)
Регистрационный № _____ Серия _____ № _____
Прицеп 1 _____ Государственный номерной знак _____ Гаражный номер
(марка)
Прицеп 2 _____ Государственный номерной знак _____ Гаражный номер
(марка)
Прицеп 3 _____ Государственный номерной знак _____ Гаражный номер
(марка)
Прицеп 4 _____ Государственный номерной знак _____ Гаражный номер
(марка)
Сопровождающие лица: _____

Работа водителя и автомобиля

операция	время по графику: число	месяц	ч.	мин.	нулевой пробег, км	показание спидометра, км	время фактическое, число, месяц, ч. мин.
1	2	3	4	5	6	7	8
выезд из гаража							
возвращение в гараж							

Движение горючего

горючее: марка	код марки	выдано, л	остаток при: выезде, л	возвращении, л	сдано, л	коэффициент изменения нормы	Время работы, ч. мин.: спецоборудования	двигателя
9	10	11	12	13	14	15	16	17

подпись

заправщика	механика	механика	заправщика	диспетчера

ЗАДАНИЕ ВОДИТЕЛЮ

в чье распоряжение (наименование и адрес заказчика)	время прибытия, ч. мин.	адрес пункта: погрузки	разгрузки	наименование груза	количество ездок	расстояние, км	перевезти тонн
18	19	20	21	22	23	24	25
				Итого			

Водительское удостоверение проверил, задание выдал,
выдать горючего _____ литров

Диспетчер _____ _____
(подпись) (расшифровка подписи)

Водитель по состоянию здоровья к управлению допущен _____ _____ _____
(должность) (подпись) (расшифровка подписи)

Место для штампа

Автомобиль технически исправен. _____
Выезд разрешен. Механик _____ _____
(подпись) (расшифровка подписи)
Автомобиль принял. Водитель _____ _____
(подпись) (расшифровка подписи)

При возвращении автомобиль исправен / неисправен
Сдал водитель _____ _____
(подпись) (расшифровка подписи)
Принял механик _____ _____
(подпись) (расшифровка подписи)

Отметки организации-владельца автотранспорта: _____

Рисунок 2.4 – Путевой лист грузового автомобиля, Форма 4-С (лицевая сторона)

Путевой лист включает в себя следующие сведения:

— сведения об организации (наименование, адрес, телефон);

— сведения о подвижном составе (марка, модель, регистрационные номера);

— сведения об экипаже (ФИО, номер водительского удостоверения и т.п.);

— сведения о движении горючего, пробеге и времени движения подвижного состава;

— подписи ответственных лиц: диспетчера, механика, водителя;

— маршрут движения подвижного состава.

То есть в путевом листе содержится сведения, которые представляет собой разные виды защищаемой информации: открытые данные (сведения о юридическом лице), персональные данные (сведения о водительских удостоверениях) и сведения, составляющие коммерческую и финансовую тайну (маршрут передвижения ТС, движения ГСМ и т.п.).

Маршрут – заранее намеченный путь следования с указанием пунктов и порядка их прохождения.

Маршруты бывают:

— маятниковые, то есть происходит многократное перемещения между двумя пунктами (отправления и назначения) и обратно (используется форма путевого листа 4-П);

— кольцевые, движение по замкнутой траектории (используется форма 4-С)[1].

Маршрут (последовательность выполнения задания) располагается на обратной стороне путевого листа, которая часто называется маршрутной (см. рисунок 2.5).

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ

пункт погрузки, разгрузки и перецепки прицепов	номер ездки	прибытие: число	прибытие: ч.	прибытие: мин.	убытие: ч.	убытие: мин.	номер прицепа: прибывших	номер прицепа: убывших	порожний пробег прицепа	номера приложенных товарно-транспортных документов (ТТД)			Наименование грузоотправителя (грузополучателя)	Подпись и печать грузоотправителя (грузополучателя)
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8													

Всего ☐ ТТД в количестве ☐ _____ шт. (прописью)

Таксировка

Сдал водителя _____ (подпись) _____ (расшифровка подписи) Принял диспетчер _____ (подпись) _____ (расшифровка подписи)

Особые отметки: _____

Простои на линии: причина наименование	причина код	дата (число, месяц), время, ч. мин. начало	дата (число, месяц), время, ч. мин. окончание	Подпись ответственного лица
41	42	43	44	45

РЕЗУЛЬТАТЫ РАБОТЫ АВТОМОБИЛЯ И ПРИЦЕПОВ

расход горючего, л: по норме	расход горючего, л: фактически	время в наряде, ч. мин.: всего: автомобиля	время в наряде, ч. мин.: всего: прицепа	в том числе автомобиля: в движении	в том числе автомобиля: в простое: всего	в простое: под погрузкой, разгрузкой: всего	в простое: под погрузкой, разгрузкой: сверхнормат.	в простое: по техническим неисправностям	количество: ездок	количество: заездов	пробег, км: общий: автомобиля	пробег, км: общий: прицепа	пробег, км: в том числе с грузом: автомобиля	пробег, км: в том числе с грузом: прицепа	перевезено, т: всего	перевезено, т: в том числе на прицепах	выполнено, ткм: всего	выполнено, ткм: в том числе на прицепах	зарплата: код	зарплата: сумма, руб. коп.
																			65	66
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64		

Коды марок: Автомобиля ☐ Прицепов ☐ ☐ ☐ Автомобиль, дни в работе ☐ Таксировщик _____ (подпись) _____ (расшифровка подписи)

Рисунок 2.5 – Путевой лист грузового автомобиля, Форма 4-С (обратная сторона)

Также на обратной стороне путевого листа располагаются сведения о результатах работы подвижного состава (расходе топлива, времени в пути, времени в простое, пробеге и т.д.)

В структуре автотранспортного предприятия (АТП) можно выделить следующие отделы и службы:

— администрация;

— техническая служба;

— отдел снабжения;

— служба эксплуатации.

Типовая организационная структура АТП показана на рисунке 2.6.

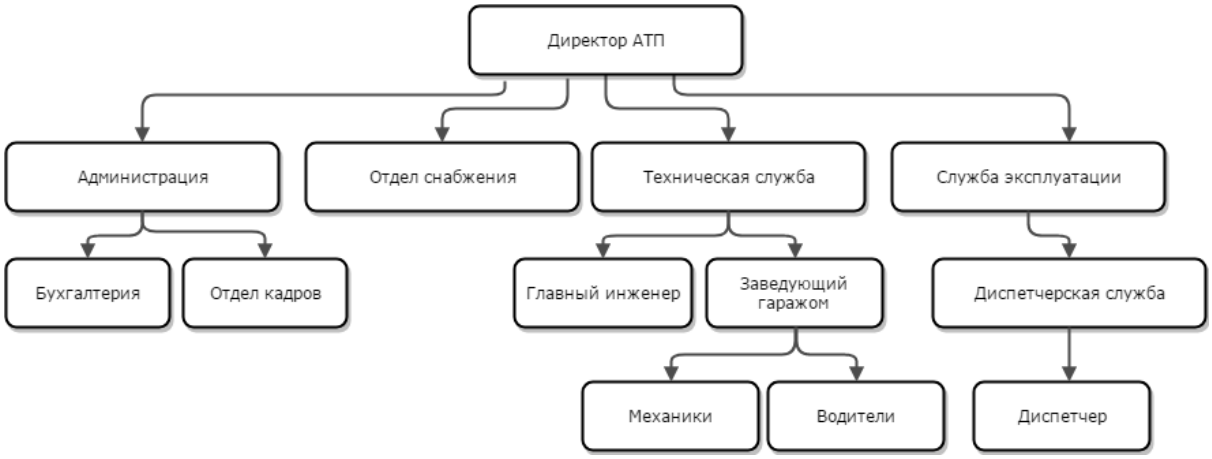


Рисунок 2.6 – Организационная структура АТП

Администрация состоит из бухгалтерии и отдела кадров. В штате бухгалтерии имеется главный бухгалтер, бухгалтера и экономисты. Отел кадров занимается подбором сотрудников, определением потребности организации в кадрах, аттестации сотрудников и т.д.

Отдел снабжения разрабатывает планы закупок и отпуска необходимых ресурсов. Снабжает всем необходимым другие структурные единицы предприятия.

Основной задачей технической службы является поддержка автопарка предприятия в исправном состоянии. Проводит плановые технические

осмотры, ремонт техники, предрейсовые осмотры и т.п. Главный инженер руководит деятельностью технической службы. Главный механик обеспечивает бесперебойную эксплуатацию подвижных составов, путем поддержания в работоспособном состоянии всех узлов транспортных средств.

Служба эксплуатации (диспетчерская служба) разрабатывает план перевозок, распределяет заявки по подвижным составам, ведет учёт анализирует перевозки.

В штате службы эксплуатации находятся диспетчеры. Диспетчер отвечает за координацию движения автотранспорта предприятия. Работа диспетчера связана с обработкой большого количества информации, поступающей к нему. Системы спутникового мониторинга позволили упростить этот труд, за счет ускорения получения информации о состоянии автотранспорта, а также автоматизации её обработки и структурирования.

Подсистема направлена на автоматизацию работы диспетчерской службы, однако её пользователями являются не только диспетчеры, а также механики, водители, бухгалтера и другие лица, причастные к оформлению путевых и маршрутных листов. Также нельзя забывать об администраторах системы, которые должны иметь доступ ко всему функционалу.

В связи с этим в подсистеме необходимо реализовать политики доступа для каждой группы пользователей, а также разграничить уровни доступа в соответствии с должностными обязанностями сотрудников.

2.1.5 Алгоритм функционирования подсистемы

Для решения задачи используется вычислительная техника. Решение задачи состоит из следующих этапов:

1. Ввод необходимых сведений об автопарке, водителях, пунктах развоза и заявках в подсистему диспетчером.

2. Распределение заявок по транспортным средствам и водителям, которые прикрепляются к автомобилю.

3. Оптимизация маршрута движения.

4. Формирование заполненного путевого листа.

Для выполнения первого этапа задачи диспетчер использует интерфейс ввода данных в подсистему.

Для решения второго этапа используется следующий алгоритм действий:

— из списка доступного транспорта выбирается автомобиль;

— определяются расстояния от начального пункта до каждого доступного пункта развоза;

— выбирается ближайший пункт и назначается данному автомобилю (данный пункт помечается как недоступный);

— определяются расстояния от выбранного пункта до каждого из доступных пунктов развоза;

— до тех пор, пока не будет достигнут предел грузоподъемности автомобиля или предел по времени движения по маршруту, который равен продолжительности смены водителя, автомобилю назначаются ближайшие пункты развоза (данные пункты помечается как недоступные).

Данный алгоритм повторяется до тех пор, пока не будут обработаны все заявки, либо пока не останется доступных автомобилей.

При решении третьего этапа по оптимизации маршрута рассматривается классическая задача коммивояжера.

Коммивояжер – разъездной торговец какой-либо фирмы, который выступает в роли посредника и предлагает товары.

Впервые задача коммивояжера была описана в 1832 году, в дальнейшем задача переросла в классическую задачу дискретной оптимизации. На примере которой было разработано большое количество различных методов оптимизации.

Задача коммивояжера (ЗК) или travelling salesman problem (TSP) заключается в поиске кратчайшего замкнутого маршрута, проходящего через заданные города хотя бы по одному разу. В случае, если ставится условие о

том, что в каждый город можно заходить только один раз, то говорится о поиске гамильтонова контура[935].

ЗК можно представить в виде графа, где вершины графа являются городами, а ребра между вершинами – дорогами между ними, вес ребра соответствует расстоянию между этими городами. Представление ЗК в виде графа показано на рисунке 2.7.

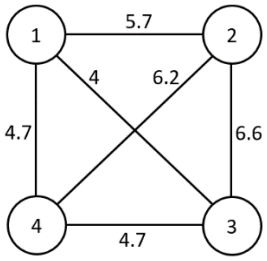


Рисунок 2.7 – Представление Задачи Коммивояжера в виде графа

Также ЗК можно представить в виде матрицы $P_{i,j}$. В данном случае строки и столбцы матрицы будут соответствовать городам, а элементы матрицы – расстоянию между соответствующими городами. Диагональные элементы матрицы, как правило, принимаются за бесконечность. Представление ЗК в виде графа показано на рисунке 2.8.

	1	2	3	4
1	∞	5,7	4	5,2
2	6,8	∞	6,6	7,9
3	4,7	4,7	∞	3,5
4	4,7	6,2	4,3	∞

Рисунок 2.8 – Представление Задачи Коммивояжера в виде матрицы

Различают симметричные и асимметричные ЗК[30]. В первом случае расстояния из любого города в любой другой город и обратно равны, во втором случае обратный путь может отличаться. При работе с реальными

задачами чаще сталкиваются именно с асимметричными задачами, т.к. дорожная сеть в современных городах является сложной структурой с применением развязок, дорог с односторонним движением и прочих решений по организации дорожного движения.

ЗК является алгоритмически сложной и относится к классу NP-трудных задач. Количество возможных маршрутов экспоненциально зависит от количества городов. Количество маршрутов для ассиметричной ЗК можно вычислить по формуле (2.1).

$$k = n!, \tag{2.1}$$

где n – количество городов, k – количество маршрутов

Для симметричной ЗК необходимо разделить полученное количество маршрутов пополам.

Существуют различные методы решения ЗК. Из которых при проектировании подсистемы по формировании путевых листов в результате анализа был выбран генетический алгоритм, как наиболее быстрый и выдающий наиболее стабильный результат.

Генетический алгоритм взял за основу естественную эволюцию, включая процессы скрещивания, естественного отбора и мутации.

Алгоритм работает с популяцией, которая содержит особи. Особи имеют определенный генотип, который состоит из хромосом. Хромосома может быть битом, числом или другим значением, с помощью которого можно оценить приспособленность генов[35].

Идея алгоритма заключается в оценке приспособленности каждой особи популяции при помощи специальной функции и отборе наиболее хорошо приспособленных особей. Отобранные особи в дальнейшем скрещиваются и составляют новую популяцию. Также введено понятие мутации, когда меняется одна или несколько хромосом гена, мутация происходит с определенной вероятностью.

Рассмотрим подробнее работу генетического алгоритма.

Алгоритм начинается с инициализации начальное популяции, как правило хромосомы в начальной популяции генерируются случайным образом. В случае решения ЗК хромосома представляет собой номер пункта маршрута, а ген содержит только уникальные значения.

После инициализации начальной популяции оценивается приспособленность популяции. Для ЗК приспособленность особи определяется длиной маршрута.

Когда стало известно о степени приспособленности особей популяции в дело вступает отбор. Одним из методов отбора является проведение турнира (турнирный отбор). Случайным образом выбирается четное количество особей, которые разделяются на две группы. В каждой группе выбираются наиболее приспособленные особи, которые в дальнейшем скрещиваются (см. рисунок 2.9).

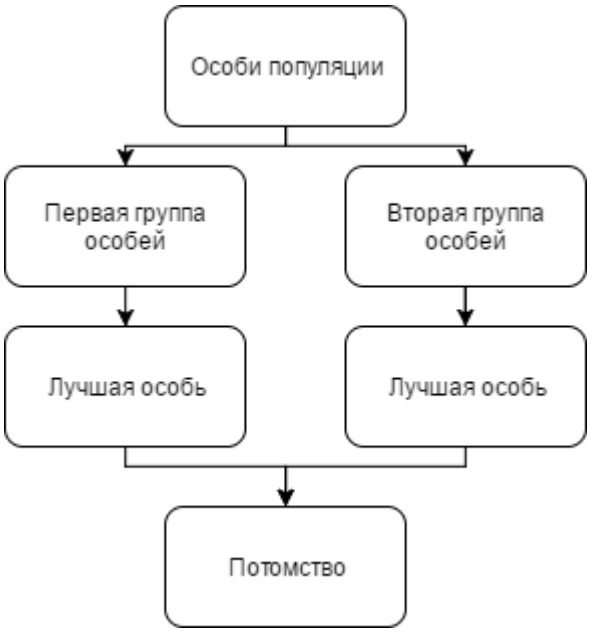


Рисунок 2.9 – Турнирный отбор

Процесс скрещивания представляет собой создание нового генотипа, я имеющего часть хромосом от первого родителя и остальную часть от второго родителя. Новый генотип называют потомком.

Скрещивание бывает двух видов: скрещивание в одной точке и скрещивание в нескольких точках[35]. Оба вида показаны на рисунке 2.10.

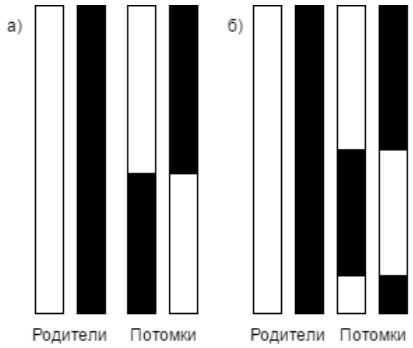


Рисунок 2.10 – Турнирный отбор: а) в одной точке; б) в нескольких точках.

Для решения ЗК был модифицирован алгоритм, т.к. необходимо сохранять уникальность хромосом в пределах одного гена. Берется подмножества из генотипа первого родителя и добавляется к генотипу потомка. Далее отсутствующие хромосомы (узлы маршрута) по порядку добавляются потомку от второго родителя, за исключением хромосом подмножества первого родителя[35]. Процесс скрещивания для ЗК представлен на рисунке 2.11.

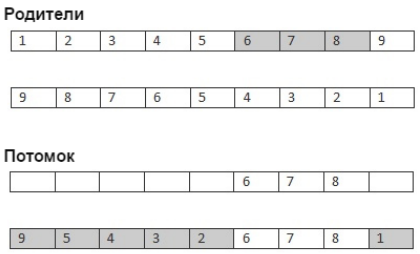


Рисунок 2.11 – Скрещивание генов

Мутация – это процесс внесения изменений в хромосомы генотипа. Мутация позволяет внести новый генетический материал в популяцию, тем самым может «встряхнуть» популяцию и предотвратить преждевременное схождение. Обычно мутация просто заменяет одну хромосому генотипа

другим значением, но для решения ЗК необходимо поменять местами две случайные хромосомы генотипа (см. рисунок 2.12).

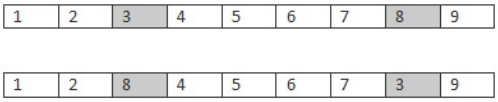


Рисунок 2.12 – Процесс мутации

Также возможно использование элитарной системы, сущность которой состоит в том, что в новую популяцию автоматом попадает определенное количество наиболее приспособленных особей.

Работа генетического алгоритма показана на рисунке 2.13.

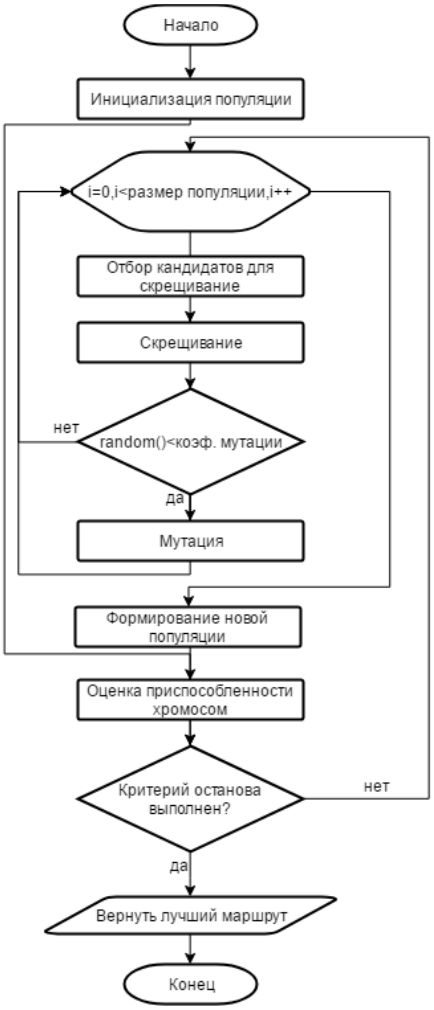


Рисунок 2.13 – Блок-схема генетического алгоритма

Критерием остановки генетического алгоритма является достижение максимального количества поколений (итераций) или отсутствие изменения лучшего значения на протяжении определенного количества итераций.

2.1.6 Информационная модель и ее описание

Информационная модель представляет собой совокупность информации, которая характеризует основные свойства и состояния объектов (сущностей) предметной области, а также взаимосвязь между ними[26].

В подсистеме по формированию путевых листов имеются следующие сущности:

— организации;
— диспетчеры клиентов;
— водители клиентов;
— график работы водителей;
— водительские удостоверения;
— транспортные средства клиентов;
— лицензионные карточки транспортных средств;
— пункты развоза;
— заявки клиентов;
— путевые листы;
— маршруты.

Физическая модель базы данных, показывающая таблицы и связи между ними, представлена на рисунке 2.14.

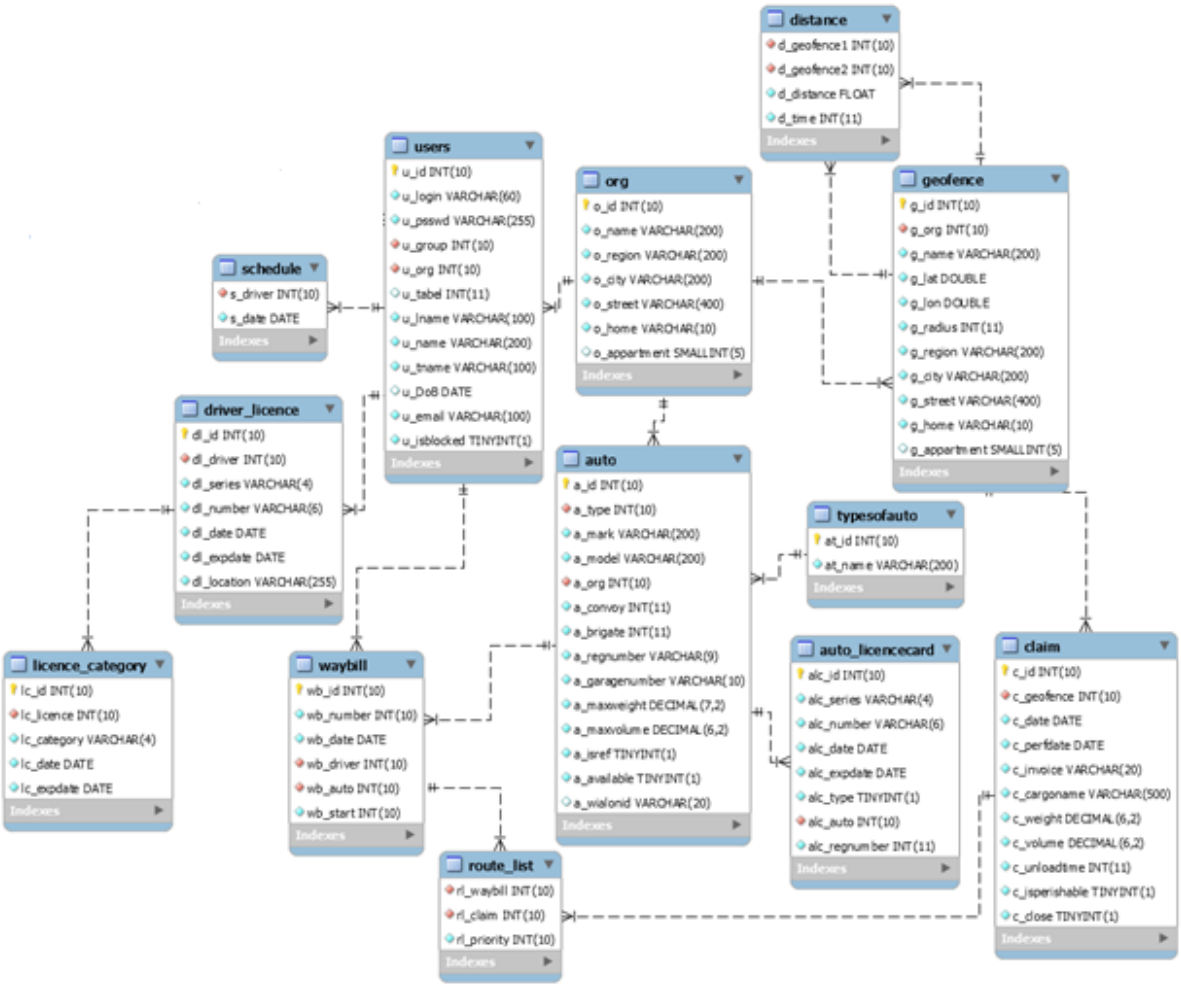


Рисунок 2.14 – Физическая модель данных

База данных подсистемы состоит из 13 таблиц, каждая из которых хранит сведения об определенной сущности, либо выступает в качестве справочников. Рассмотрим подробнее каждую из таблиц БД.

Таблица «org» (Организации) хранит необходимые для заполнения путевых листов сведения об организациях, которые являются клиентами оператора услуг и пользуются услугами подсистемы (см. таблицу 2.4).

Таблица 2.4 – Структура таблицы «org»

Название	Тип данных	Атрибуты	Описание поля
o_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле

Продолжение таблицы 2.4

Название	Тип данных	Атрибуты	Описание поля
o_name	Текстовый		Наименование организации
o_region	Текстовый		Регион
o_city	Текстовый		Название города
o_street	Текстовый		Название улица
o_home	Текстовый		Номер дома или строения
o_appartment	Числовой		Номер офиса

Таблица «users» (Пользователи) хранит сведения о пользователях подсистемы. Пользователями подсистемы сотрудники оператора, предоставляющего услуги, которые выделены в группу «Администраторы», диспетчеры клиентов (группа «Диспетчеры») и водители клиентов(группа «Водители»).

Таблица пользователей содержит поля со сведениями о дате рождения и табельном номере, но так как эти сведения важны только для пользователей, состоящих в группе «Водители», то для этих полей был установлен атрибут «Allow Null», который позволяет оставлять эти поля пустыми. Структура таблицы показана в таблице 2.5.

Таблица 2.5 – Структура таблицы «users»

Название	Тип данных	Атрибуты	Описание поля
u_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле

Продолжение таблицы 2.5

Название	Тип данных	Атрибуты	Описание поля
u_login	Текстовый	Уникальный	Имя пользователя, использующееся для входа в подсистему
u_passwd	Текстовый		Хранит хэш пароля пользователя
u_group	Текстовый	Беззнаковый	Указывает на принадлежность пользователя к одной из групп
u_org	Числовой	Беззнаковый	Организация пользователя, внешний ключ, ссылается на таблицу «org»
u_tabel	Числовой	Null	Табельный номер
u_lname	Текстовый		Фамилия
u_name	Текстовый		Имя
u_tname	Текстовый		Отчество
u_DoB	Дата	Null	Дата рождения
u_email	Текстовый		E-mail пользователя
u_isblocked	Логический		Статус блокировки

Таблица «schedule» (График работы) хранит сведения о графике работы водителей клиентов (см. таблицу 2.6). Таблица 2.6 – Структура таблицы «schedule»

Название	Тип данных	Атрибуты	Описание поля
s_driver	Числовой	Беззнаковый	Водитель, внешний ключ, ссылается на таблицу «users»
s_date	Дата		Дата

В таблице «schedule» введено ограничение на уникальность значений полей s_driver и s_date. Для этого был создан составной ключ, включающий данные поля и установлен атрибут «Уникальный» для созданного составного ключа.

Таблица «driver_licence» (Водительские удостоверения) хранит сведения, необходимые для заполнения путевых листов, о водительских удостоверениях водителей клиентов (см. таблицу 2.7).

Таблица 2.7 – Структура таблицы «driver_licence»

Название	Тип данных	Атрибуты	Описание поля
dl_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
dl_driver	Числовой	Беззнаковый	Водитель, внешний ключ, ссылается на таблицу «users»
dl_series	Текстовый		Серия вод. удостоверения
dl_number	Числовой		Номер вод. удостоверения
dl_date	Дата		Дата выдачи вод. удостоверения
dl_expdate	Дата		Дата окончания действия вод. удостоверения
dl_location	Текстовый		Место выдачи вод. удостоверения

Таблица «licence_category» (Категории водительских удостоверений) хранит сведения, необходимые для заполнения путевых листов, об открытых категориях водительских удостоверений водителей клиентов (см. таблицу 2.8).

Таблица 2.8 – Структура таблицы «licence_category»

Название	Тип данных	Атрибуты	Описание поля
lc_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
lc _licence	Числовой	Беззнаковый	Водитель, внешний ключ, ссылается на таблицу «users»
lc_category	Текстовый		Серия водительского удостоверения
lc _date	Дата		Дата выдачи водительского удостоверения
lc _expdate	Дата		Дата окончания срока действия категории водительского удостоверения

Таблица «typesofauto» (Типы автомобилей) хранит название типа и его идентификатор (см. таблицу 2.9).

Таблица 2.9 – Структура таблицы «groups»

Название	Тип данных	Атрибуты	Описание поля
at_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
at_name	Текстовый		Название типа

Таблица «auto» (Автомобили) хранит сведения, необходимые для заполнения путевых листов, об автомобилях клиентов (см. таблицу 2.10).

Таблица 2.10 – Структура таблицы «auto»

Название	Тип данных	Атрибуты	Описание поля
a_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
a_type	Числовой		Тип авто, внешний ключ, ссылается на таблицу «typesofauto»
a_mark	Текстовый		Марка автомобиля
a_model	Текстовый		Модель
a_org	Числовой	Беззнаковый	Организация, внешний ключ, ссылается на таблицу «org»
a_convoy	Числовой	Беззнаковый	Номер колонны, которой принадлежит автомобиль
a_brigate	Числовой	Беззнаковый	Номер бригады, которой принадлежит автомобиль
a_regnumber	Текстовый		Регистрационный номер
a_garagenumber	Числовой		Гаражный номер
a_maxweight	Числовой		Максимальная грузоподъемность
a_maxvolume	Числовой		Объем кузова
a_isref	Логический		Наличие рефрижератора
a_available	Логический		Статус доступности
a_wialonid	Текстовый		Идентификатор автомобиля в системе спутникового мониторинга

Таблица «auto_licncecard» (Лицензионная карточка автомобиля) хранит сведения, необходимые для заполнения путевых листов, об открытых лицензионных карточках автомобилей клиентов (см. таблицу 2.11).

Таблица 2.11 – Структура таблицы «auto_licncecard»

Название	Тип данных	Атрибуты	Описание поля
alc_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
alc _series	Текстовый		Серия лиц. карточки
alc_number	Текстовый		Номер лиц. карточки
alc _date	Дата		Дата выдачи лицензионной карточки
alc _expdate	Дата		Дата окончания срока действия лиц. карточки
alc_type	Логический		Тип лицензионной карточки
alc_auto	Числовой	Беззнаковый	Автомобиль, внешний ключ, ссылается на таблицу «auto»
alc_regnumber	Текстовый		Номер лицензии

Таблица «geofence» (Пункты развоза) хранит необходимые для заполнения путевых листов сведения о пунктах развоза клиентов, а также сведения о географических координатах для интеграции с системами мониторинга и решения задачи коммивояжера (см. таблицу 2.12).

Таблица 2.12 – Структура таблицы «geofence»

Название	Тип данных	Атрибуты	Описание поля
g_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле

Продолжение таблицы 2.12

Название	Тип данных	Атрибуты	Описание поля
g_org	Числовой	Беззнаковый	Организация, внешний ключ, ссылается на таблицу «org»
g _name	Текстовый		Наименование организации
g_lat	Числовой		Долгота
g_lon	Числовой		Широта
g_radius	Числовой		Радиус
g _region	Текстовый		Регион
g _city	Текстовый		Название города
g _street	Текстовый		Название улица
g _home	Текстовый		Номер дома или строения
g _appartment	Числовой		Номер офиса

Таблица «distance» (Расстояния) хранит расстояния и время между пунктами развоза клиентов (см. таблицу 2.13). Данные сведения необходимы для решения задачи коммивояжера.

Таблица 2.13 – Структура таблицы «distance»

Название	Тип данных	Атрибуты	Описание поля
d_geofence1	Числовой	Беззнаковый	Первый пункт развоза, внешний ключ, ссылается на таблицу «geofence»
d_geofence2	Числовой	Беззнаковый	Второй пункт развоза, внешний ключ, ссылается на таблицу «geofence»
d_distance	Числовой		Растояние межу пунктами
d_time	Числовой		Время между пунктами (сек.)

В таблице «distance» введено ограничение на уникальность значений полей d_geofence1 и d_geofence2. Для этого был создан составной ключ, включающий данные поля и установлен атрибут «Уникальный» для него.

Таблица «claim» (Заявки) хранит необходимые для заполнения путевых листов сведения о заявках клиентов (см. таблицу 2.14).

В таблице «claim» введено ограничение на уникальность значений полей c_geofence и c_perfdate. Для этого был создан составной ключ, включающий данные поля и установлен атрибут «Уникальный» для него.

Таблица 2.14 – Структура таблицы «claim»

Название	Тип данных	Атрибуты	Описание поля
c_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
c_geofence	Числовой	Беззнаковый	Пункт доставки, внешний ключ, ссылается на таблицу «geofence»
c_date	Дата		Дата поступления заявки
c_perfdate	Дата		Требуемая дата выполнения
c_invoice	Текстовый		Номер товарной накладной
c_cargoname	Текстовый		Название груза
c_weight	Числовой		Масса груза
c_volume	Числовой		Объем груза
c_unloadtime	Числовой		Время разгрузки в секундах
c_isperishable	Логический		Указывает на то, что груз является скоропортящимся
c_close	Логический		Статус заявки

Таблица «waybill» (Путевые листы) хранит о сформированных путевых листах (см. таблицу 2.15).

Таблица 2.15 – Структура таблицы «waybill»

Название	Тип данных	Атрибуты	Описание поля
wb_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
wb_date	Числовой		Дата путевого листа
wb_number	Числовой		Номер путевого листа
wb_driver	Числовой	Беззнаковый	Водитель, внешний ключ, ссылается на таблицу «users»
wb_auto	Числовой	Беззнаковый	Автомобиль, внешний ключ, ссылается на таблицу «auto»
wb_start	Числовой		Пункт погрузки

Таблица «route_list» (Маршруты) хранит о маршрутах сформированных путевых листах (см. таблицу 2.16).

Таблица 2.16 – Структура таблицы «route_list»

Название	Тип данных	Атрибуты	Описание поля
rl_waybill	Числовой	Беззнаковый	Путевой лист, внешний ключ, ссылается на таблицу «waybill»
rl_claim	Числовой	Беззнаковый	Заявка (пункт погрузки), внешний ключ, ссылается на таблицу «claim»
rl_priority	Числовой		Порядковый номер пункта маршрута

2.1.7 Структура подсистемы

При анализе подсистемы были выявлены функции, которые выполняются пользователями для решения поставленной задачи, а также для обеспечения ввода необходимой информации в базу данных, что позволит определить критичный Выявленные функции показаны в виде схемы на рисунке 2.15.

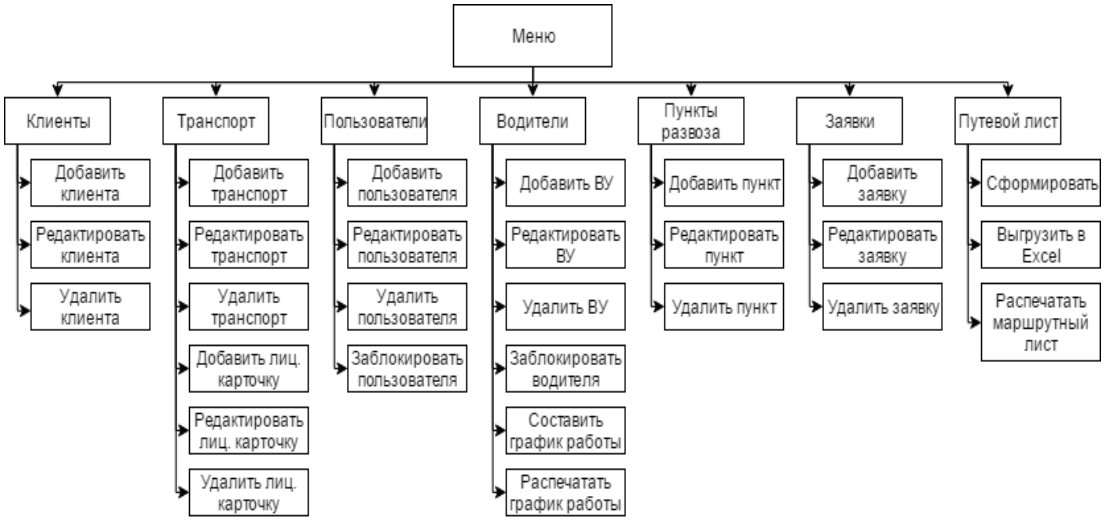


Рисунок 2.15 – Дерево функций подсистемы

Подсистема состоит из набора модулей. Каждый модуль представляет собой PHP сценарий и выполняет определённые функции. Также в структуре подсистемы содержаться HTML формы и файлы сценариев JavaScript. Структура модулей и форм подсистемы представлена на рисунке 2.16.

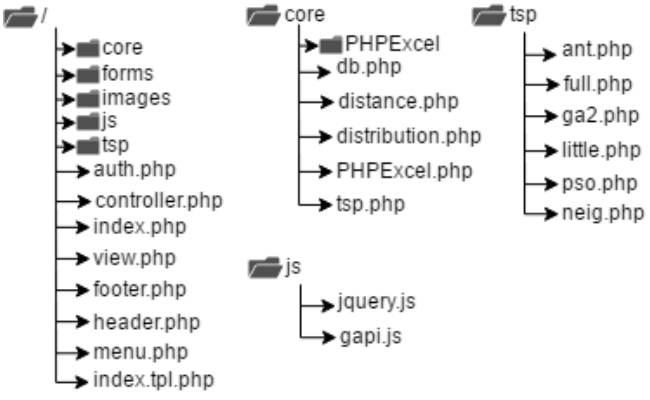


Рисунок 2.16 – Структура подсистемы

К основным модулям подсистемы относятся:

— auth.php – модуль авторизации;

— index.php – непосредственно к этому модулю обращается пользователь через HTTP запросы, модуль предназначен для вывода шаблона index.form.tpl;

— view.php – формирует представления данных, полученных из БД;

— controller.php – обеспечивает контроль данных, поступающих в БД;

— header.php – шаблон шапки интерфейса;

— footer.php – шаблон подвала интерфейса;

— menu.php – шаблон главного меню.

Модули ядра располагаются в папке «core». К ним относятся:

— db.php – выполняет все запросы, поступившие от пользователя;

— distance.php – возвращает информацию о расстоянии между двумя пунктами, используя Google Maps Distance Api;

— distribution.php – модуль, распределяющий заявки по транспортным средствам;

— tsp.php – модуль, решающий задачу коммивояжера генетическим алгоритмом;

— PHPExcel.php – свободно распространяемая библиотека, предоставляющая широкие возможности по работе с файлами Microsoft Excel (xls, xlsx).

Папка «tsp» содержит реализации методов решения задачи коммивояжера.

В папке «images» содержится графическая информация, использующаяся в интерфейсе подсистемы.

Папка «forms» содержит формы разных разделов сайта, а также шаблон путевого листа в формате Microsoft Excel.

Папка «js» содержит библиотеку jQuery и модуль gapi.js, предназначенный для работы с API Google Maps.

Архитектура подсистемы основана на применении шаблонов проектирования: Router (Маршрутизатор) и MVC (Model-View-Controller).

Шаблон «Router» позволяет скрыть элементы системы, путем сведения всех внешних запросов к одному объекту[4]. На рисунке 2.17 представлена схема, показывающая модель взаимодействия системы с пользователем, при использовании данного шаблона проектирования.

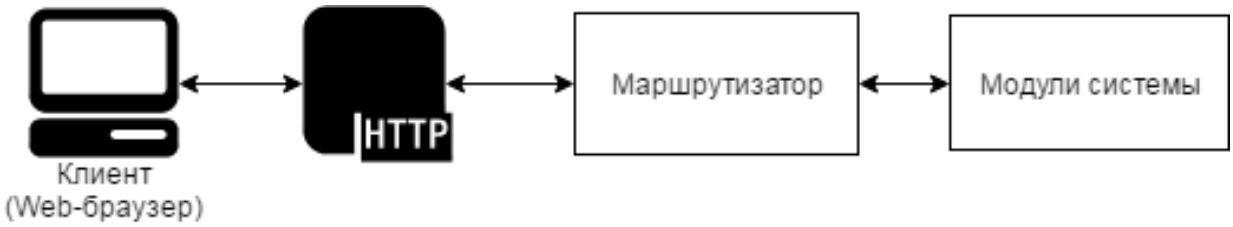


Рисунок 2.17 – Шаблон проектирования «Router»

Шаблон MVC состоит из трех отдельных компонентов:

— Model – предоставляет доступ к данным и методы работы с ними;

— View – формирует представление данных, полученных из модели, и выводит их клиенту;

— Controller – обрабатывает запросы пользователей[4].

На рисунке 2.18 представлена схема, показывающая модель взаимодействия системы с пользователем, при использовании данного шаблона проектирования.

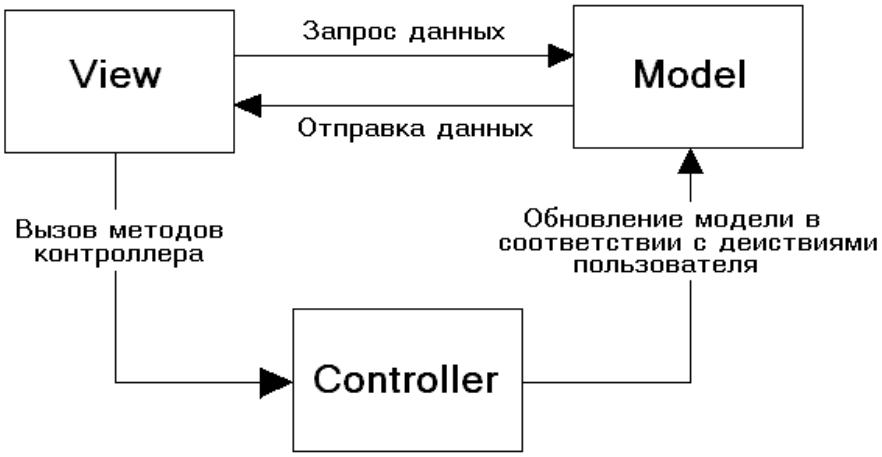


Рисунок 2.18 – Шаблон проектирования «MVC»

Схема взаимодействия программных модулей и информационных файлов представляет собой гибрид рассмотренных шаблонов проектирования и отражает связи программного и информационного обеспечения подсистемы и представлена на рисунке 2.19.

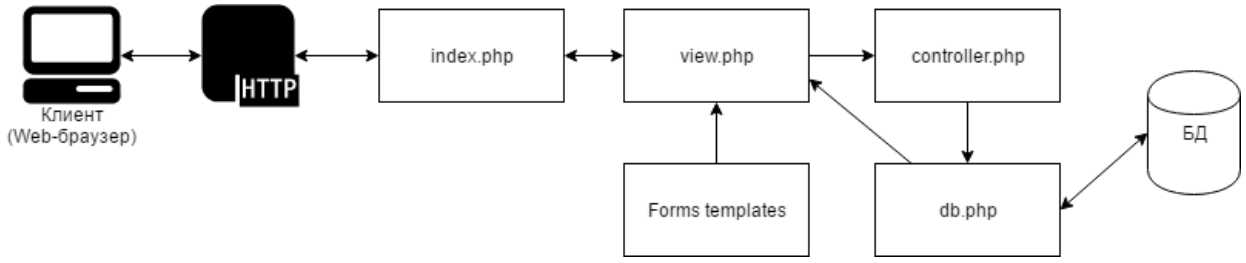


Рисунок 2.19 – Схема взаимодействия модулей подсистемы

Из данной схемы видно, что запросы, поступающие от пользователей (клиентов) передаются на сервер по протоколу HTTP. Клиент сценарий index.php загружает шаблон index.tpl.php, а также содержимое раздела подсистемы (страницы), полученное в ответе сценария view.php. Для обеспечения перехода по разделам сайта без перезагрузки страницы используется технология AJAX.

AJAX – это акроним, который раскрывается как Asynchronous JavaScript and XML (асинхронный Javascript и XML), заключается в способе обмена данными между сервером и клиентом в фоновом, при котором ответ сервера

передается в формате HTML, XML или JSON и обрабатываются клиентом без необходимости перезагрузки страницы[7]. Использование данной технологии уменьшить нагрузку на сервер, а также сократить трафик, благодаря меньшему объему передаваемых данных.

Модуль view.php взаимодействует с модулем controller.php, который проверяет корректность запроса, поступившего от пользователя и направляет его к модулю db.php. Модуль db.php предназначен для взаимодействия с базой данных MySQL. В качестве провайдера данных используется PDO (PHP Data Object).

PDO – является интерфейсом для доступа к базе данных. Его отличительной особенностью является абстрактность, то есть в независимости от того, какая база данных используется функции для выполнения запросов останутся без изменений[620].

После обработки запроса db.php возвращает результат модулю view.php, который формирует представление данных и выводит его в интерфейс пользователя.

В дальнейшем были рассмотрены программные модули подсистемы, выполняющие основные функции.

Модуль auth.php используется для авторизации пользователей, а также на каждой странице сайта для проверки сессии. Используется встроенный в PHP механизм сессий, который заключается в присваивании каждому пользователю уникального идентификатора, который используется в дальнейшем при аутентификации пользователей в процессе работы с подсистемой.

Сессия начинается через вызов функции session_start(), после чего можно устанавливать или считывать переменные сессии, которые хранятся в массиве «$_SESSION»[6]. При завершении работы с сессией вызывается функция session_write_close().

Модуль авторизации включается в каждую страницу посредством функции include, и в случае отсутствия корректной сессии перенаправляет пользователя на форму авторизации.

Модуль index.php загружает шаблон index.tpl.php, шапку интерфейса (header.php), подвал (footer.php) и меню (menu.php). Модуль включает в себя скрипт, позволяющий выполнять AJAX запросы. Также подключается библиотека jQuery и gapi.js.

Gapi.js предназначен для работы с Google Maps Java Script API. Google Maps API – это набор готовых компонентов для работы с картами Google и внедрения их в собственные приложения[36].

Googel Maps API существует для разных платформ, в том числе Android и iOS[36]. Также существует версия для веб-браузера и JavaScript. Обращение к картографическому сервису от Google происходит посредством HTTP запросов.

Компания Google предоставляет ряд служб, реализуемых через API:

— Geocoding API - обеспечивает геокодирование адресов;

— Directions API позволяет прокладывать маршруты между разными точками;

— Distance Matrix API позволяет получить информацию о расстоянии и времени поездки между двумя точками;

— Elevation API предоставляет сведения о высоте;

— Geolocation API возвращает местоположение устройства;

— Places API предоставляет информацию о различных местах;

— Roads API позволяет находить дорогу и определить ограничения скорости на ней;

— Time Zone API возвращает сведения о часовом поясе в точке, заданной географическими координатами[33].

Модуль «gapi.js» содержит функции инициализации карты, создания маркеров на карте, инициализации службы автоматической подстановки адреса по геоданным,

Для реализации данного функционала использовались следующие классы Google Maps JavaScript API:

— google.maps.Map() – создает карту и помещает её в div-контейнер;

— google.maps.places.Autocomplete() – создает поле автоматической подстановки адреса;

— google.maps.Marker() – создает маркер на карте для наглядного отображения местоположения объектов[1233].

Ядро подсистемы составляют важнейшие модули подсистемы, обеспечивающие.

Модуль «db.php» создает объект данных PHP (PDO), с которым в дальнейшем можно работать в любом месте сценария. Функция «run» предназначена для выполнения SQL запроса и принимает два аргумента: строку SQL запроса и массив параметров, которые используются в запросе. Особенностью PDO является использование подготовленных запросов, в которые передаются параметры, что повышает безопасность приложения[20].

Модуль «distance.php» предназначен для получения матрицы расстояний и времени пути между пунктами и занесением их в базу данных подсистемы. Вызывается при добавлении нового пункта развоза или при обновлении существующего для того, чтобы в базе данных всегда существовали все расстояния.

Для получения матрицы расстояний используется Google Maps Distance Matrix API. Точки отправления и назначения передаются посредством GET запросов, после чего API возвращает ответ в виде JSON.

Используется функция PHP file_get_contents() для получения ответа и функция json_decode() для десериализации JSON в ассоциативный массив. В дальнейшем из этого получаются значения полей «distance» и «time», которые вносятся в таблицу «distance» базы данных.

Модуль «distribution.php» содержит алгоритм распределения заявок по автомобилям. Содержит основную функцию «distribute», которой в параметрах передаётся идентификатор клиента, дата, и стартовый пункт погрузки. И функцию «getDistance», которая извлекает из базы матрицу расстояний от стартового пункта ко всем пунктам организации.

В основной функции происходит выбор всех заявок клиента на определенную дату, в дальнейшем выбирается все доступные транспортные средства клиента и водители, работающие в этот день.

Каждому автомобилю назначаются заявки, до тех пор, пока не будет достигнуть предел грузоподъемности автомобиля, либо предел по вместимости кузова, либо пока не будет достигнут предел по времени работы водителя.

Результатом работы подсистемы является: заполненный путевой лист грузового автомобиля по форме 4-С;.

После распределения заявок по транспортным средствам диспетчер может выгрузить путевой лист в файл. В качестве формата файла был выбран формат Microsoft Excel, т. к. существует большое количество средств для просмотра и редактирования данного формата.

Путевой лист заполняется следующими данными:

— сведениями об автомобиле;

— сведениями о водителе;

— сведениями о лицензии;

— сведениями о задании (маршруте).

Так как в путевом листе содержатся сведения о пробеге транспортного средства, времени выезда и возвращения на базу, а также результаты работы водителя и транспортного средства, которые должны заполняться водителем по факту, а также заверяться подписью должностных лиц, то эти сведения не заполняются подсистемой.

Маршрутный лист содержит порядок прохождения пунктов маршрута с указанием адреса и номера товарной накладной. Используется водителем как вспомогательная информация.

Помимо этого, для водителя предоставляется ссылка на картографический сервис Google Maps, перейдя по которой водитель может посмотреть маршрут движения транспортного средства на карте, а также воспользоваться приложением Google Maps в качестве навигатора на смартфоне.

2.2 Постановка задачи проектирования технологии обеспечения информационной безопасности корпоративного WEB-приложения

2.2.1 Нормативная база

При разработке технического задания на систему безопасности рекомендуется руководствоваться требованиями и рекомендациями документов ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»[10] и ГОСТ 51583-2000 «Защита информации. Порядок создания автоматизированных систем в защищенном исполнении»[11]:

— общие сведения;

— назначение и цели создания системы защиты;

— характеристика объекта;

— требования к системе;

— состав и содержание работ по созданию системы защиты;

— порядок контроля и приемки системы защиты;

— требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы защиты в действие;

— требования к документированию.

2.2.2 Модель нарушителя информационной безопасности корпоративного WEB-приложения

Модель нарушителя – это формализованное абстрактное описание потенциального злоумышленника, деятельность которого может быть направлена на реализацию каких-либо угроз безопасности для совершения противоправных действий с информационными активами организации.

Модель нарушителя включает в себя:

— Цели и мотивы нарушителя;

— Тип нарушителя;

— Степень подготовленности и оснащенности;

— Описание используемых средств;

— Описание конкретных способов реализации угрозы;

— Механизмы сокрытия следов;

— Способы защиты от действий нарушителя.

Разработаем модель нарушителя для типового WEB-приложения.

Для начала определим цель. Целью нарушителя будет получение, уничтожение или искажение ценных сведений, хранящихся в приложении и используемых при работе с ним.

Степень подготовленности нарушителя зависит от типа реализуемой угрозы и может быть как минимальной (в случае реализации непреднамеренных угроз или сговора), так и очень высокой (в случае реализации атак на веб-сервер). В качестве оснащения будут выступать инструментальные средства анализа WEB-приложений (сканеры портов, сканеры уязвимостей и т.п.).

В качестве нарушителя может выступать как сотрудник организации, пользователь приложения, так и внешний нарушитель, действующий в своих интересах или интересах конкурентов.

Сценарии действия нарушителя как правило связаны с поиском уязвимых мест во взаимодействии клиентской части приложения с WEB-сервером. После обнаружения уязвимости наступает фаза реализации угроз, после которой злоумышленник может попытаться скрыть факт проникновения путем удаления и изменения журналов действий.

2.2.3 Цель и назначение разрабатываемой подсистемы

Подсистема предназначена для повышения эффективности транспортных перевозок, за счет автоматизации заполнения путевого листа грузового автомобиля по форме 4-С, а также сокращения пробега транспортных средств, следующих по кольцевому маршруту, за счет использования алгоритмов поиска кратчайшего пути. Предназначена для использования диспетчерами службы эксплуатации автотранспортных предприятий.

Система защиты подсистемы создается с целью:

— исключение и затруднение получения злоумышленником защищаемой информации;

— сокращения рисков, связанных с утечкой информации из подсистемы[2].

2.2.4 Общая характеристика организации решаемых задач

Первостепенная задача обеспечения информационной безопасности в подсистеме по формированию путевых листов – обеспечение доступности, конфиденциальности и целостность информации. Проблема доступности решена организацией за счет реализации систем резервирования. Для решения

задач по обеспечению конфиденциальности и целостности применяются механизмы аутентификации, мониторинга и аудита.

Аутентификация – это процесс проверки подлинности пользователя. В случае успешной аутентификации пользователь получает авторизацию, то есть правда на доступ к определенным ресурсам[3].

Существуют различные методы аутентификации:

— Парольная аутентификация (пользователь обладает некоторой уникальной информацией, которая используется при входе в систему);

— Аутентификация с применением электронной подписи;

— Аутентификация с применением аппаратных устройств;

— Биометрическая аутентификация.

В настоящий момент в подсистеме используется аутентификация по многоразовому паролю. Но в случае кражи пароля злоумышленником, он может получить доступ к функционалу подсистемы.

Для решения подобного рода проблем безопасности применяется подход с использованием нескольких факторов аутентификации (многофакторная аутентификация).

В ходе анализа работы пользователей с подсистемой было выявлено что использование аппаратных средств и биометрических данных не всегда является возможным, так как вход может осуществлять с разных устройств в разных местах, а также не все пользователи подсистемы готовы к финансовым затратам на приобретение дополнительного оборудования. Поэтому в качестве второго фактора аутентификации было принято решение использовать одноразовый пароль с ограниченным временем действия.

Алгоритмы генерации одноразовых паролей:

— HOTP – генерация пароля на учете прошлых аутентификаций пользователя. В алгоритме при генерации одноразового пароля используется количество авторизаций. Каждый следующий пароль будет уникален. Основным недостатком является возможное рассогласование счетчиков на сервере и устройстве[24];

— TOTP – алгоритм генерации, основанный на использовании времени. Его достоинством является ограниченное время существования сгенерированного пароля;

— OCRA – модификация алгоритмов TOTP и HOTP. Его основным отличием является то, что в качестве аргумента для генерации пароля используется случайное число, принятое от сервера, то есть добавляется цепочка запроса и ответа между ними.

Пароль может генерироваться как независимо от сервера, в таком случае используются специальные устройства или приложения, так и на сервере, в таком случае пользователи могут получать одноразовые пароли по разным каналам, но чаще всего используется SMS или E-mail.

Существуют программные решения для генерации одноразовых паролей, такие как:

— Google Authenticator – приложение от Google который поддерживает генерацию по методам HOTP и TOTP$

— Yandex Key;

— Microsoft Authentificator.

Но их использование может быть осложнено политикой предоставления услуг, которые могут ограничивать сферу применений, а также необходимостью настройки приложений на устройствах пользователей.

Поэтому для реализуемой в рамках данной выпускной квалификационной работы подсистеме защиты был выбран метод генерации одноразового пароля на стороне сервера, основанный на времени и передача кода пользователю посредством СМС-сообщения, так как у компании имеется вся необходимая инфраструктура для реализации.

На этапе проектирования подсистемы формирования путевых листов было установлено что сессия пользователя хранится только в рамках текущего сеанса. Это компромисс между удобством пользования и приемлемым уровнем безопасности. Реализация сессий пользователей основана на применении встроенных методов PHP.

Вторым этапом повышения уровня безопасности является внедрение системы протоколирования сессий и действий пользователя.

2.2.5 Обоснование проектных решений

Техническое обеспечение (ТО) – это совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных информационных системах[27].

Для обеспечения функционирования подсистемы используется сервер со следующими характеристиками:

— процессор Intel Core i7;

— оперативная память 8гб или больше;

— 2 дисковых накопителя емкостью 100гб или больше.

Для обеспечения бесперебойного доступа к подсистеме необходима установка источника бесперебойного питания.

Необходимо обеспечить систему резервного копирования, с использованием внешних накопителей и/или серверов резервного копирования. Помимо этого, дисковые накопители, используемые в сервере, должны быть объединены в массив RAID1.

Необходимо организовать доступ сервера в сеть интернет, для обеспечения клиентов доступом к подсистеме.

Оператор системы мониторинга уже располагает сервером с доступом в интернет, необходимой сетевой инфраструктурой, источником бесперебойного питания, а также системой резервного копирования, то есть приобретение дополнительного оборудования не требуется.

Информационное обеспечение представляет собой совокупность входных и выходных документов, методов их построения, состава классификаторов, способ организации информационной базы и т.п.

Входными данными в подсистеме являются:

— сведения об автотранспорте (марка, модель, регистрационный номер, гаражный номер, грузоподъемность, тип, номер колонны и бригады, которым принадлежит автомобиль, сведения о лицензионной карточке);

— сведения о водителе (ФИО, дата рождения, табельный номер, серия и номер водительского удостоверения, открытые категории водительского удостоверения)

— сведения об пунктах развоза (адрес, географические координаты, название);

— сведения о заявках (пункт развоза, требуемая дата, номер товарной накладной, наименование груза, масса);

— сведения о графике работы водителей.

Выходными данными являются:

— сведения о распределении автопарка по заявкам;

— сведения о маршруте (порядке прохождения заявок).

На выходе подсистема позволяет сформировать путевой лист по форме 4-С, заполненный необходимыми сведениями.

При выборе программного обеспечения учитывалась специфика подсистемы. Подсистема представляет собой WEB-приложение.

WEB-приложение – это программное обеспечение с клиент-серверной архитектурой, где в качестве клиента выступает web-браузер пользователя, а взаимодействие с сервером организуется посредством HTTP-запросов. Для обработки запросов используется WEB-сервер[29].

WEB-сервер принимает HTTP запросы и выдает HTTP-ответы на них.

Соответственно для функционирования подсистемы необходим WEB-сервер, СУБД, серверный и клиентский язык программирования.

Рассмотрим наиболее распространённое ПО WEB-сервера: nginx, Apache и Microsoft IIS.

Apache – свободное, кроссплатформенное ПО, к основным достоинствам относится гибкость и надежность. Имеет большое количество

модулей, которые добавляют поддержку различных языков программирования. Получил широкое распространение по всему миру.

Nginx – свободный web-сервер, разработанный российским программистом в 2002 году. Поддерживает большое количество операционных систем. Отличается быстротой и надежностью. Основной особенностью nginx является его скорость работы со статическим контентом, за счет использования механизма неблокирующих соединений, однако для работы с динамическим контентом он подходит меньше и может выступать в роли прокси-сервера.

Microsoft IIS (Internet Information Service) – проприетарный web-сервер, функционирующий на операционных системах семейства Microsoft Windows. Ориентирован на обеспечение надежности и безопасности.

Так как в компании уже располагается сервер с установленной операционной системой GNU/Linux, то использование WEB-сервера от Microsoft не представлялось возможным. Так как подсистема подразумевает выдачу большого количества динамического контента, то в качестве WEB-сервера был выбран Apache.

В качестве клиентского скриптового языка программирования используется JavaScript, так как поддерживается всеми современными браузерами, в том числе и мобильными[19].

Язык разметки страницы – HTML и CSS, в качестве языка описания стилей документа.

В качестве языка программирования используется PHP.

PHP (PHP HyperText Processor) – скриптовый, интерпретируемый язык программирования, интенсивно применяющийся при разработке web-приложений[17].

Выбор PHP обосновывается рядом факторов:

— PHP является свободным программным обеспечением;

— поддерживается большим сообществом разработчиков пользователей;

— постоянно развивается;

— имеет множество сторонних библиотек, расширяющих базовый функционал;

— поддерживает большое количество разнообразных платформ, про положительно сказывается на переносимости web-приложения при смене хостинга или сервера.

В качестве системы управления БД используется свободная реляционная СУБД MySQL. В качестве инструмента для администрирования базой данных MySQL использовался проект с открытым исходным кодом phpMyAdmin.

При разработке подсистемы для создания и редактирования скриптов PHP, шаблонов HTML и прочих файлов проекта использовался свободный текстовый редактор Notepad++.

Выводы по разделу.

В данной главе была проанализирована деятельность Оператора системы мониторинга, изучена его структура, программное и техническое обеспечение. Дана характеристика используемой информационной подсистемы формирования путевых листов грузового автомобиля. И обоснована необходимость разработки подсистемы защиты информации. Были рассмотрены методы решения поставленной задачи. Также были рассмотрены существующие программные продукты для решения аналогичных задач. Проведены обоснования проектных решений по различным аспектам.

3 ПРОЕКТНАЯ ЧАСТЬ

3.1 Информационное обеспечение модуля безопасности корпоративного WEB-приложения

3.1.1 Инфологическая модель после модернизации

Для реализации поставленных задач в базу данных были добавлены следующие сущности:

— сессии пользователей;

— действия пользователей;

— группы пользователей;

— разрешения.

Таблица «sessions» (Сессии пользователей) хранит сведения, необходимые для определения информации о сессии пользователя (см. таблицу 3.1).

Таблица 3.1 – Структура таблицы «sessions»

Название	Тип данных	Атрибуты	Описание поля
ses_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
ses_sid	Текстовый	Беззнаковый, null	Идентификатор сессии, выданный скриптом PHP при авторизации
ses_user	Числовой	Беззнаковый	Пользователь, внешний ключ, ссылается на таблицу «users»
ses_ip	Текстовый		IP-адрес, с которого был произведен вход в систему
ses_datetime	Дата/Время		Дата и время входа в систему
ses_expdate	Дата/Время		Дата и время конца действия сессии

Продолжение таблицы 3.1

Название	Тип данных	Атрибуты	Описание поля
ses_success	Логический		Успешность входа
ses_error	Числовой		Код ошибки

Поле «ses_success» используется для протоколирования неудачных попыток входа. Регистрируются все неудачные попытки входа с указанием кода ошибки в поле «ses_error»: 1 – неверно введенный пароль, 2 – неверно введен одноразовый пароль. Если вход был успешен, то код ошибки – 0 и поле «ses_success» равно «true».

В таблице «user_logs» (Журнал действий пользователей) записывается информация о действиях пользователей (см. таблицу 3.2).

Таблица 3.2 – Структура таблицы «user_logs»

Название	Тип данных	Атрибуты	Описание поля
log_sid	Числовой	Беззнаковый	Сессия пользователя, внешний ключ, ссылается на таблицу «sessions»
log_section	Текстовый		Раздел сайта
log_action	Числовой	Беззнаковый	Действие пользователя
log_table	Текстовый		Таблица базы данных
log_datetime	Дата/Время		Дата и время выполнения действия
log_objectID	Числовой	Беззнаковый	Идентификатор объекта
log_success	Логический		Успешность выполнения операции
log_error	Текстовы		Описание ошибки

Поле «log_success» позволяет выявить ошибки в работе подсистемы, а также попытки эксплуатации уязвимостей. Поле «log_error» хранит тестовое представление ошибки, которое может вернуть база данных в случае некорректного запроса.

Таблица «groups» (Группы) хранит название группы и её идентификатор (см. таблицу 3.3).

Таблица 3.3 – Структура таблицы «groups»

Название	Тип данных	Атрибуты	Описание поля
g_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
g_name	Текстовый		Название группы

Таблица «permission» (Разрешения) хранит сведения о правах доступа разных групп пользователей к разделам подсистемы (см. таблицу 3.4).

Таблица 3.4 – Структура таблицы «permission»

Название	Тип данных	Атрибуты	Описание поля
p_id	Числовой	Беззнаковый, уникальный, счетчик	Ключевое поле
p_group	Числовой	Беззнаковый	Группа пользователя, внешний ключ, ссылается на таблицу «groups»
p_section	Текстовый		Раздел сайта
p_permission	Числовой		Уровень доступа (от 0 до 7)

Также для функционирования системы двухфакторной аутентификации была модифицирована таблица «Users», в которую было добавлено текстовое поле «u_phone».

Физическая модель базы данных после модификации показана на рисунке 3.1.

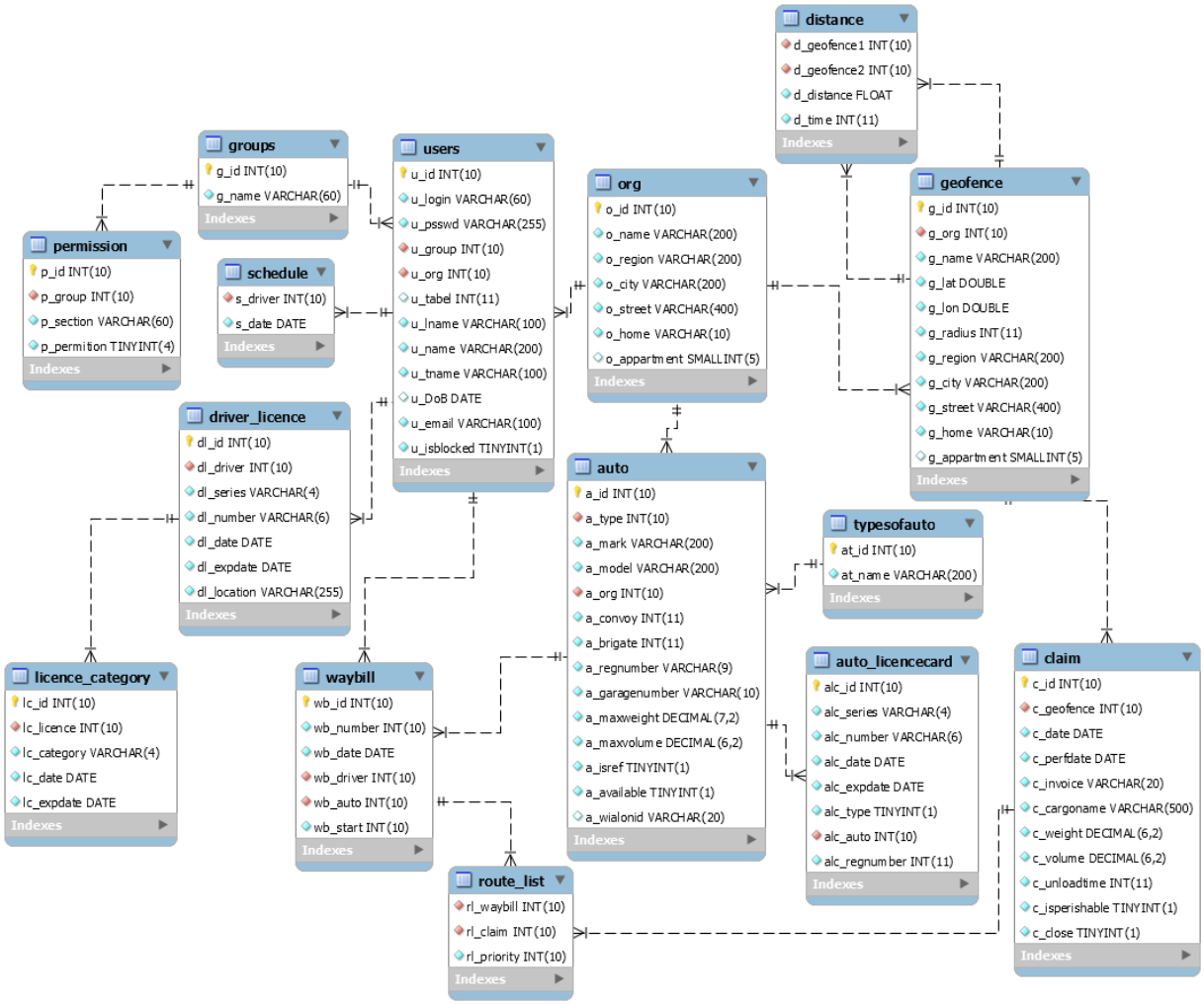


Рисунок 3.1 – Физическая модель данных

3.1.2 Структура подсистемы после модернизации

Помимо инфологической модели модернизации подверглись модули подсистемы. Были добавлены библиотеки генерации одноразового пароля и взаимодействия с телематическим сервером («TOTP.php» и «wialon.php»).

Также модификации подверглись модули авторизации и работы с базой данных и была добавлена директория «logs», в которой хранятся пользовательские журналы.

Структура подсистемы после модернизации показана на рисунке 3.2

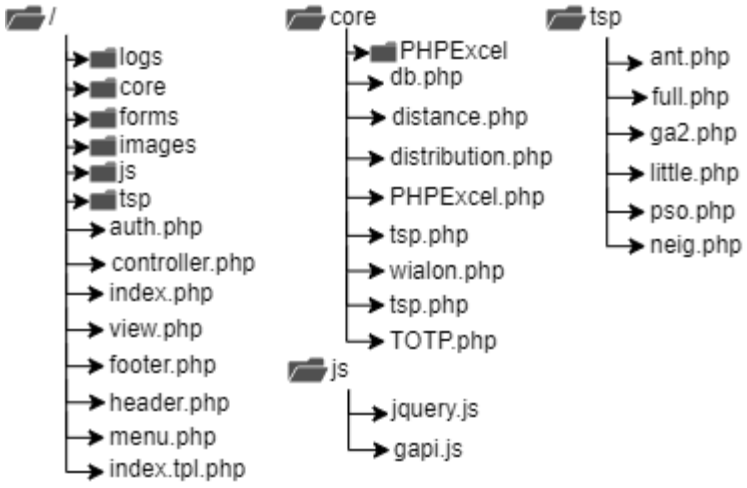


Рисунок 3.2 – Структура подсистемы после модернизации

3.2 Разработка технологии обеспечения модуля безопасности корпоративного веб-приложения

3.2.1 Организация взаимодействия с телематическим сервером

В процессе обновления функционала подсистемы возникла необходимость обеспечения безопасного взаимодействия с телематическим сервером предприятия. Для решения этой задачи используется стандарт REST API, входящий в состав Wialon SDK[13].

Для работы с методами REST API используется модуль wialon.php, который подключается в необходимых модулях и создается Wialon API.

Аутентификация на сервере происходит с применением стандарта JSON Web Token. Он позволяет повысить уровень безопасности при взаимодействии подсистем с внешними сервисами, за счет отсутствия прямого обращения к серверу по логину и паролю, которые заменены на токен. Токен проверяется на сервере секретным ключом, также время действия токена ограничено и

имеется возможность обновить токен вручную, в случае возникновения подозрений в его утечке.

Листинг кода взаимодействия с телематическим сервером на примере авторизации:

```
include('wialon.php'); //подключение библиотеки
$wialon_api = new Wialon(); //создание объекта класса взаимодействия с телематическим сервером
$token = 'Token'; //Токен, полученный на сервере
$result = $wialon_api->login($token); //результат авторизации в формате JSON
$json = json_decode($result, true); //десоциализация результата
if(!isset($json['error'])){
    //код при успешном выполнении входа
} else {
    echo WialonError::error($json['error']); //вывод кода ошибки при неуспешном выполнении входа
}
```

3.2.2 Модуль двухфакторной авторизации

Для генерации паролей был выбран метод TOTP. Способ доставки пароля - СМС. Был модифицирован модуль авторизации.

Принципиальная схема взаимодействия программных модулей в процессе двухфакторной авторизации показана на рисунке 3.3

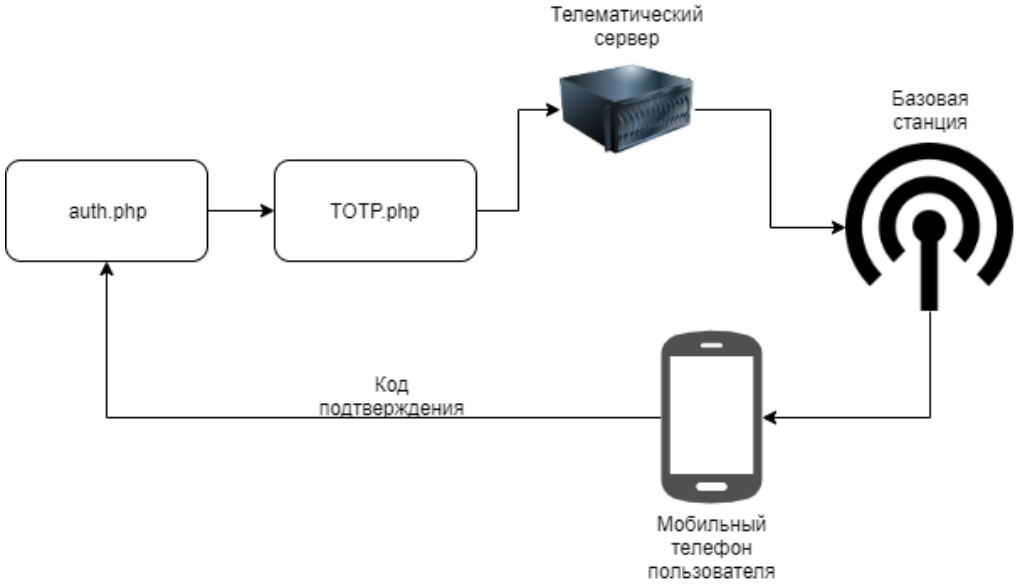


Рисунок 3.3 – Схема функционирования двухфакторной авторизации

При входе в подсистему пользователю показывается форма входа, показанная на рисунке 3.4

Рисунок 3.4 – Форма входа

После успешного прохождения проверки логина и пароля происходит генерация одноразового кода модулем «TOTP.php» и отправка СМС сообщения.

Для отправки СМС сообщения используется телематический сервер системы спутникового мониторинга. Так как он поддерживает отправку СМС, то достаточно подключиться к нему, используя SDK и отправить код при помощи команды «svc=user/send_sms». Данная команда принимает два параметра: «phoneNumber» - номер телефона в тестовом формате и «smsText» – текст сообщения[13].

В коде модуля этот запросы выглядит следующим образом:

$wialon_api->user_ send_sms ('{" phoneNumber ":'+phone+',"smsText":Код подтверждения: '+$totpCode+'});

После отправки СМС сообщения пользователь вводит его в форму, показанную на рисунке 3.5.

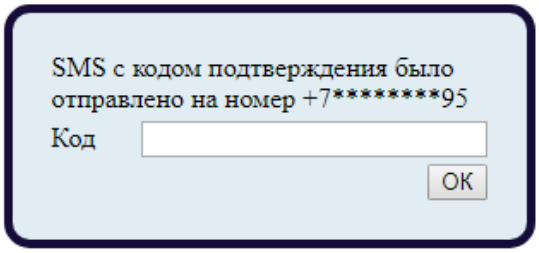


Рисунок 3.5 – Форма входа (Ввод одноразового пароля)

После успешного прохождения второго фактора проверки механизмами PHP производится генерация сессии.

3.2.3 Реализация механизма протоколирования действий пользователей

Так как архитектура приложения представляет собой паттерн MVC, то все обращения (чтение, изменение, создание и удаление) к базе данных происходят через модуль db.php. Поэтому для создания журнала действий пользователей достаточно включить в модуль функцию, которая будет записывать всю необходимую информацию об операциях пользователей.

Функция создает запись в базе данных при каждом обращении пользователя к ней. Записи добавляются в таблицу «user_logs» и содержат следующую информацию:

— Идентификатор сессии пользователя, который позволяет определить какой пользователь совершил данную операцию;

— Раздел сайта, в котором совершалось действие;

— Действие (символ, показывающий операцию с данными: «+» - создание записи, «-» - удаление записи, «*» - обновление записи);

— Таблицу, к которой был совершен пользовательский запрос;

— Идентификатор объекта таблицы базы данных;

— Время выполнения операции;

— Текст ошибки (если имеется).

Также помимо записи действий в базу данных реализовано протоколирование в текстовый файл.

Для записи в файл используется функция «file_put_contents»[5]. Журналы действий пользователей хранятся в директории «logs» и разделены по датам для уменьшения объема файлов и упрощения работы с файлами журналов.

В коде модуля запись в файлы выглядит следующим образом:

file_put_contents('./logs/userlog_'.date("j.n.Y").'.log',$log, FILE_APPEND);

3.3 Технологическое обеспечение модуля безопасности корпоративного веб-приложения

3.3.1 Описание интерфейса подсистемы после модернизации

Интерфейс подсистемы изначально представляет собой динамический WEB-интерфейс. Пользователи приложения работают с подсистемой, используя WEB -браузер. Меню состоит из следующих разделов:

— Клиенты;

— Транспорт;

— Пользователи;

— Водители;

— Пункты развоза;

— Заявки;

— График работы;

— Путевые листы.

В рамках разработки модуля в подсистему был добавлен еще один пункт меню – «Администрирование» (см. рисунок 3.6).

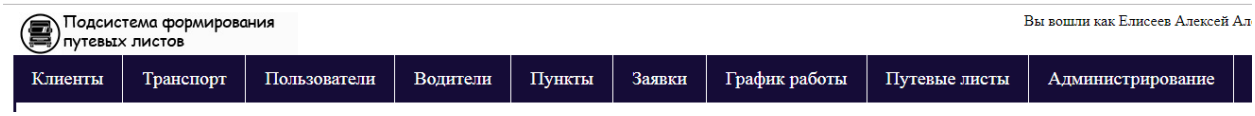


Рисунок 3.6 – Меню подсистемы

Раздел «Администрирование» включает в себя инструменты администратора для просмотра журналов входов и действий пользователей. Интерфейс раздела показан на рисунке 3.7.

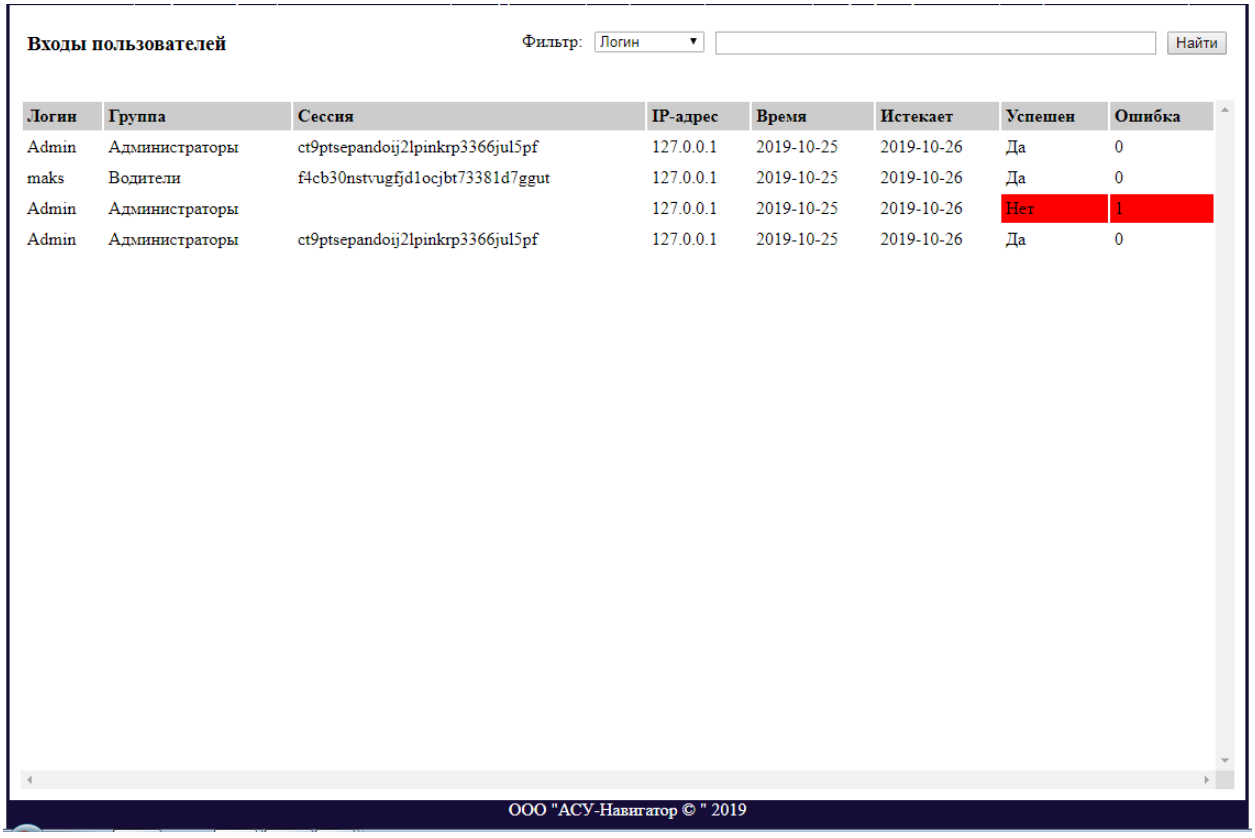


Рисунок 3.7 – Интерфейс администратора подсистемы

Для того чтобы администратору было проще обращать внимания на опасные с точки зрения информационной безопасности события они выделяются в списке цветовой схемой.

Как и каждый раздел подсистемы, раздел администратора предусматривает возможность фильтрации выдаваемой информации по различным полям. Пример формы фильтрации представлен на рисунке 3.8.

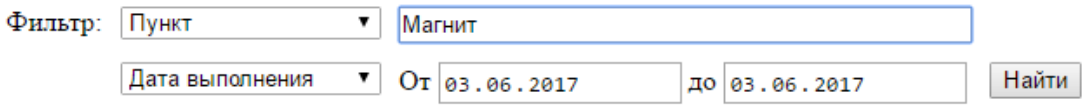


Рисунок 3.8 – Форма фильтрации

Фильтрация содержимого производится с помощью SQL оператора LIKE.

3.3.2 Технология обеспечения разграничения доступа пользователей

Разработана система разграничения доступа, которая реализуется за счет групп пользователей и разрешений, дающих определенные привилегии пользователям при доступе к разделам подсистемы.

В структуре базы данных выделены таблицы «group» и «permission». Каждый пользователь подсистемы принадлежит определенной группе, а каждая группа имеет определенные права на доступ к каждому разделу сайт.

Права доступа к разделу определяются числом, длинной в три бита. Каждый бит которого дает доступ к различному функционалу подсистемы. Первый бит – разрешает добавление записей, второй бит – редактирование и удалений и третий бит – дает право просмотра. Если все три бита числа равны нулю, то пользователь не имеет прав доступа к разделу.

В базе данных уровень привилегий хранится в виде десятичного числа в пределах от 0 до 7. Такой способ удобен для человека и для хранения в базе данных, но не удобен при распределении прав доступа в скриптах подсистемы. Поэтому прежде всего данное число преобразуется в строку, состоящую из

стрех символов, каждый из которых имеет значение 0 или 1. Данное преобразование осуществляется при помощи функции decbin().

Администраторы подсистемы имеют доступ ко всем разделам и объектам. Диспетчеры имеют доступ только к объектам своей организации, а также не могут просматривать информацию о других клиентах. Водители имеют доступ только к разделу путевых листов и могут просматривать только путевые листы и маршруты, выписанные на их имя.

3.3.3 Реализация методов защиты от прочих видов атак

В процессе анализа исходного кода подсистемы по формированию путевых листов были выявлены недостатки в работе с вводимыми данными пользователей. При разработке были реализованы методы защиты от SQL-инъекций, заключающиеся в подготовке запроса методами PDO, который исключает передачу вредоносного кода вместе с параметрами запроса, но при этом в некоторых формах имеется возможность реализации XSS атак.

Для защиты от XSS применяются встроенных функций экранирования входных и выходных значений. В PHP к таким функциям относятся[18]:

— «strip_tags»;

— «htmlemtities»;

— «htmlspecialchars».

Данные функции экранируют данные, которые передаются в них в качестве аргумента. Под экранированием понимается что любые спецсимволы, влияющие на выполнение кода или запроса путем подстановки символа обратной косой черты – «\».

Также был выявлен недостаток в заголовках страниц: не на всех страницах была указана используемая кодировка. Это серьезный недостаток проектирования приложения, так как в случае, если кодировка указана неверно, то браузер сам определяет её, и злоумышленник может использовать

форму для встраивания вредоносного кода в другой кодировке, обойдя этап экранирования при отправке формы.

Для решения данной проблемы безопасности достаточно добавить в заголовок страницы тэг <meta charset="Encoding">, где под «Encoding» понимается требуемая кодировка, используемая в приложении.

В конфигурации PHP-сервера было рекомендовано включить директиву «session.cookie_httponly», которая делает клиентские файлы Cookie недоступными для скриптов на стороне клиента.

Еще одной серьезной проблемой безопасности стало отсутствие шифрования в использованном протоколе передачи данных. Для решения этой задачи был осуществлен переход на протокол HTTPS, который основан на применении ассиметричной схемы шифрования (то есть с применением открытого и закрытого ключей для установления соединения) и симметричного шифрования для передачи данных после установления соединения[31].

Для использования протокола HTTPS необходимо сгенерировать и установить сертификаты открытого и закрытого ключа. Можно использовать так называемые «самоподписанные» сертификаты, но это не обеспечивает достаточного уровня безопасности при использовании публичных сетей, так как возможна реализация атаки «Человек посередине». Поэтому необходимо заказывать генерацию сертификата у специализированных удостоверяющих центров, с помощью которых в дальнейшем проверяться подлинность и актуальность сертификатов.

Для нужд подсистемы был выбран удостоверяющий центр «Let’s Encrypt», в котором бесплатно можно сгенерировать сертификаты.

После генерации ключей был сконфигурирован веб-сервер Apache на использование протокола HTTPS, для этого:

— Необходимо загрузить файлы сертификатов в директорию сервера;

— Активировать использование SSL командой «a2enmod ssl»;

— Включить использование HTTPS по умолчанию командой «a2ensite default-ssl»;

— Затем в файлах конфигурации указать путь к сертификатам для используемого домена$

— Перезагрузить веб-сервер Apache.

Также желательно отключить возможность доступа к сайту по HTTP, для этого нужно в файлах «.htaccess» указать перенаправление для всех страниц с HTTP на HTTPS.

В дальнейшем компании рекомендовано расширить систему защиты, путем приобретения средства анализа траффика WEB-приложений WAF.

Выводы по разделу.

В разделе была описана инфологическая модель базы данных, описаны сущности и связи между ними. Разработан улучшенный механизм авторизации пользователей, инструменты администратора для мониторинга опасных действий в подсистеме. Для всех разработанных и модифицированных модулей подсистемы дано описание принципов функционирования и применяемых методов.

ЗАКЛЮЧЕНИЕ

На основе проведенных иссследований, теоретических изысканий и анализа деятельности, структуры, технического и информационного обеспечения рассматриваемого WEB-приложения была показана необходимость защиты корпоративных WEB-приложений.

Были рассмотрены классы информационных систем, возможные угрозы и возможности реализации атак, дана классификация методов обеспчеения защиты информации, произведен анализ существующих средств, технологий и решений в области обеспечения информационной безопасности корпоративных WEB-приложений.

В ходевыполнения работы были реализованы методы усиления безопасности аутентификации пользователей, инструменты для администраторов системы, проведены меры по улучшению защиты на уровне исходного кода приложения, а также даны рекомендации системным администраторам.

В результате выполнения выпускной квалификационной работы модернизирована подсистема, что позволило удовлетворить потребности пользователей системы в использовании безопасного продукта, в котором риски информационной безопасности сведены к минимуму. Также в дальнейшем имеется возможность применение данных методов защиты на других подсистемах компании.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Антонов В. Н. Организация автомобильных перевозок и безопасности движения : конспект лекций/ В. Н. Антонов – Казань – 2013, –83с.
2. Артемов А.В. Информационная безопасность. Орел: МАБИВ, 2014. 256 с.
3. Васильков, А. В. Безопасность и управление доступом в информационных системах / А.В. Васильков, И.А. Васильков. - М.: Форум, 2015. - 368 с.
4. Вора П. Шаблоны проектирования веб-приложений / П. Вора – М.: Эксмо, 2011, – 870с..
5. Гейн А.А. Web-программирование на PHP. М.: ИНТУИТ.РУ «Интернет-университет информационных технологий». То же [Электронный ресурс]. - URL: http://www.intuit.ru/department/internet/phpwebprog, 2016.
6. Горев А. Э. Организация автомобильных перевозок и безопасности движения : учеб. Пособие / А.Э. Горев, Е.М. Олещенко – 3-е изд.,стер. – М.: Издательский центр «Академия», 2009, – 256с.
7. Дари К. AJAX и PHP: разработка динамических веб-приложений / К. Дари, Б. Бринзаре, Ф. Черчез-Тоза, М. Бусика; пер. с англ. Киселева А. – Спб.:Символ-плюс, 2009, – 336с.
8. Диго С. М. Проектирование и использование баз данных: Учебник. - М.: Финансы и статистика, 2016.
9. Джонс М. Т. Программирование искусственного интеллекта в приложениях / М. Тим Джонс; пер. с англ. Осипов А.И. – М.:ДМК Пресс, 2015. – 312с. [63-68,82-83]
10. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
11. ГОСТ 51583-2000. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении.

12. Документация Google Maps API [Электронный ресурс] URL: https://developers.google.com/maps/ (дата обращения: 15.05.2017).
13. Документация Wialon Pro [Электронный ресурс] URL: https://docs.wialon.com/ru/ (дата обращения: 25.05.2017).
14. Емельянова Ю. Г., Фраленко В. П. Анализ проблем и перспективы создания интеллектуальной системы обнаружения и предотвращения сетевых атак на облачные вычисления [Электронный ресурс] // Программные системы: теория и приложения. 2011. № 4 (8). С. 17-31. http://psta.psiras.ru/read/psta2011_4_17-31.pdf (дата обращения: 10.12.2017).
15. Иванова Г.С. Технология программирования. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2018. - 336 с.
16. Инькова Н.А. Современные интернет-технологии в коммерческой деятельности. - М.: Издательство «Омега-Л», 2015. - 188 с.
17. Кузнецов М. В., Симдянов И. В., Голышев С. В.PHP 5. Практика разработки Web-сайтов. — СПб.: БХВ-Петербург, 2009. — 960 с: ил.
18. Коггзол, Джон PHP. Полное руководство.: Пер. С англ. – М. – Издательский дом «Вильямс», 2008. – 752 с.
19. Лещев Д. Создание интерактивного Web-сайта.- СПб.:Питер. 2016.-544.
20. Объекты данных PHP. Документация [Электронный ресурс] http://php.net/manual/ru/book.pdo.php (дата обращения: 06.06.2017).
21. Олифер В. Компьютерные сети. Приниципы, технологии, протоколы: учебник для вузов / В. Олифер, Н. Олифер – Спб.:Питер, 2016, – 992с.
22. Олифер В. Безопасность компьютерных сетей /В. Олифер, Н. Олифер - Москва : Горячая линия-Телеком,2017, - 644с.
23. Орглтри Т. Firewalls. Практическое применение межсетевых экранов / Т. В. Оглтри – М: «ДМК Пресс», 400с.
24. Панасенко С.П. Смарт-карты и информационная безопасность / С.П. Панасенко, К. Я. Мытник – М: «ДМК Пресс», 2018, 516с.

25. Пьюривал С. Основы разработки веб-приложений / С. Пьюривал – СПб.: Питер, 2015, – 272с.
26. Райордан Р. Основы реляционных баз данных/Пер, с англ. — М.: Издательско-торговый дом «Русская Редакция», 2011. — 384 с.: ил.
27. Сердюк В.А. Организация и технологии защиты информации. Обнаружение и предотвращение информационных атак в автоматизированных системах предприятий — Москва: ВШЭ, 2011.576 с.
28. Смирнов С. Н. Безопасность систем баз данных. / С.Н. Смирнов — М: «Гелиос АРВ», 2017, 352с.
29. Томсон Лаура Разработка Web-приложений на PHP и MySQL: Пер. с англ./Лаура Томсон, Люк Веллинг. — 2-е изд., испр. — СПб: ООО «ДиаСофтЮП», 2013. — 672 с.
30. Хэмди А. Введение в исследование операций: пер. с англ./А. Хэмди. - 7-е издание – М.: Издательский дом «Вильямс», 2005. – 912с.
31. Чипига, А. Ф. Информационная безопасность автоматизированных систем / А.Ф. Чипига. - М.: Гелиос АРВ, 2013. - 336 с.
32. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей / В.Ф. Шаньгин. - М.: Форум, Инфра-М, 2017. - 416 с.
33. Dincer A. Google Maps JavaScript API CookBook / A. Dincer, B. Uraz Birmingham: Packt Publishing Ltd., 2013, 299p.
34. OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks. https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf.pdf (дата обращения: 10.10.2019)
35. Potvin J. Genetic algorithms for the traveling salesman problem. Annals of Operations Research, 1996, 63, 339-370.
36. Svennerberg G. Beginning Google Maps API 3 / G. Svennerberg – NY:Apress, 2010, - 328p.

ПРИЛОЖЕНИЕ А

Листинг программных модулей

Листинг SQL кода для обновления таблиц в базе данных:

```
CREATE TABLE sessions(
    ses_id INT ANSIGNED NOT BULL UNIQUE AUTO_INCREMENT,
    ses_user INT UNSIGNED NOT NULL,
    ses_sid VARCHAR(255),
    ses_ip VARCHAR(255),
    ses_datetime DATETIME DEFAULT CURRENT_TIMESTAMP,
    ses_expdate DATETIME,
    ses_success TINYINT(1),
    ses_error TINYINT,
    PRIMARY KEY (ses_id),
    FOREIGN KEY (ses_user) REFERENCES users (u_id)
    )
CREATE TABLE user_logs(
    log_sid INT UNSIGNED NOT NULL,
    log_section VARCHAR(255),
    log_action VARCHAR(1),
    log_table VARCHAR(255),
    log_datetime DATETIME DEFAULT CURRENT_TIMESTAMP,
    log_success TINYINT(1),
    log_error TEXT,
    FOREIGN KEY (log_sid) REFERENCES sessions (ses_id)
    )
```