

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры полностью)

09.03.03 «Прикладная информатика»

(код и наименование направления подготовки/специальности)

Корпоративные информационные системы

(направленность (профиль)/специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка веб-приложения для аналитики событий системы
контроля и управления доступом»

Обучающийся

Д.Б. Поверина

(И. О. Фамилия)

(личная подпись)

Руководитель

Е.А. Ерофеева

Тольятти 2023

Аннотация

Тема бакалаврской работы – «Разработка веб-приложения для аналитики событий системы контроля и управления доступом».

Выбор данной темы обусловлен важностью проведения анализа событий системы контроля и управления доступом (СКУД) на устройствах, работающих с различными операционными системами.

Объектом исследования являются события в базе системы контроля и управления доступом.

Предметом исследования – аналитика событий системы контроля и управления доступом.

Целью выпускной квалификационной работы является разработка веб-приложения для аналитики событий СКУД.

Методы исследования – методы и технологии проектирования веб-приложений.

Практическое значение данной бакалаврской работы заключается в создании веб-приложения, которое обеспечивает возможность анализа событий СКУД на любых устройствах, подключенных к интернету.

Это исследование состоит из введения, трех глав, заключения и списка использованной литературы. Бакалаврская диссертация имеет объем в 45 страницы текста, содержит 16 иллюстраций, 3 таблицы и базируется на 22 источниках информации.

Abstract

The topic of the bachelor's thesis is "Development of a Web Application for Event Analytics in Access Control and Management Systems". The choice of this topic is justified by the importance of conducting event analysis in access control and management systems (ACMS) on devices running various operating systems.

The object of the research is the events stored in the database of the access control and management system.

The subject of the research is the event analytics in the access control and management system.

The aim of this bachelor's thesis is to develop a web application for event analytics in ACMS.

The research methods employed include web-application design methods and technologies.

The practical significance of this bachelor's thesis lies in the creation of a web application that enables event analysis in ACMS on any internet-connected devices.

The thesis consists of an introduction, three chapters, conclusion, and a list of references. It spans 45 pages of text, includes 16 figures, 3 tables, and is based on 22 information sources.

Оглавление

1	Анализ предметной области автоматизации и постановка задачи на разработку веб-приложения для аналитики событий СКУД	7
1.1	Характеристика деятельности отдела безопасности ТГУ	7
1.2	Анализ бизнес-процесса аналитики событий СКУД.....	9
1.3	Постановка задачи на разработку веб-приложения.....	13
	для аналитики событий СКУД	13
1.4	Обзор и анализ аналогов веб-приложений для аналитики событий СКУД	15
2	Логическое проектирование веб-приложения для аналитики событий СКУД	19
2.1	Выбор технологии логического моделирования веб-приложения	19
2.2	Построение логической модели веб-приложения	20
2.3	Требования к аппаратно-программному обеспечению веб-приложения	26
3	Реализация и тестирование веб-приложения для аналитики событий СКУД	28
3.1	Выбор инструментов разработки веб-приложения	28
3.2	Разработка архитектуры веб-приложения.....	29
3.3	Разработка физической модели данных веб-приложения.....	32
3.4	Разработка веб-приложения для аналитики событий СКУД	33
3.5	Тестирование веб-приложения	39
	Заключение	42
	Список используемой литературы	43

Введение

Сейчас ТГУ применяет систему APACS 3000 для контроля и управления доступом. Эта система выполняет несколько функций, включая учет рабочего времени, контроль доступа и контроль пребывания на объекте. Ее интерфейс оформлен в стиле стандартных приложений для работы на операционной системе Microsoft Windows. Однако недостатком является невозможность использования системы для анализа событий на других операционных системах и устройствах, таких как планшеты или смартфоны.

Разработка веб-приложения позволит получать доступ к событиям из любого места в сети. Важным преимуществом такого веб-приложения является независимость от конкретной операционной системы.

Объектом исследования являются события в базе системы контроля и управления доступом.

Предметом исследования аналитика событий системы контроля и управления доступом.

Целью выпускной квалификационной работы является разработка веб-приложения для аналитики событий СКУД.

Для достижения цели выпускной квалификационной работы необходимо решить следующие задачи:

- провести анализ предметной области;
- выполнить концептуальное моделирование предметной области;
- провести анализ существующих разработок;
- разработать требования к веб-приложению;
- построить логическую модель веб-приложения;
- разработать веб-приложение аналитики событий СКУД;
- протестировать работоспособность веб-приложения.

Методы исследования – методы и технологии проектирования веб-приложений.

Практическая значимость бакалаврской работы заключается в разработке веб-приложения, позволяющего проводить аналитику событий СКУД на любых устройствах, имеющих подключение к интернету.

Данная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

Первая глава посвящена анализу предметной области и постановке задачи на разработку веб-приложения для аналитики событий системы контроля и управления доступом.

Вторая глава посвящена проектированию веб-приложения для аналитики событий системы контроля и управления доступом.

В третьей главе представлена реализация проектных решений веб-приложения для аналитики событий системы контроля и управления доступом и проведено его тестирование.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Бакалаврская работа состоит из 45 страниц текста, 16 рисунков, 3 таблицы и 22 источников.

1 Анализ предметной области автоматизации и постановка задачи на разработку веб-приложения для аналитики событий СКУД

1.1 Характеристика деятельности отдела безопасности ТГУ

Отдел информационной безопасности является структурным подразделением ФГБОУ ВО «Тольяттинский государственный университет».

Деятельность Отдела направлена:

– на исключение или существенное снижение негативных последствий (ущерба) в отношении ТГУ вследствие нарушения функционирования информационных систем, информационно-телекоммуникационных сетей и автоматизированных систем управления в результате реализации угроз безопасности информации;

– на обеспечение конфиденциальности информации, доступ к которой ограничен в соответствии с законодательством Российской Федерации;

– на повышение защищенности ТГУ от возможного нанесения ему материального, репутационного или иного ущерба посредством случайного или преднамеренного несанкционированного вмешательства в процесс функционирования информационных систем или несанкционированного доступа к циркулирующей в них информации и ее несанкционированного использования;

– на обеспечение надежности и эффективности функционирования и безопасности информационных систем, производственных процессов и информационно-технологической инфраструктуры ТГУ;

– на обеспечение выполнения требований по информационной безопасности при создании и функционировании информационных систем ТГУ.[10]

Отдел информационной безопасности имеет следующую структуру:

- начальник отдела информационной безопасности;
- инженер по технической защите информации;

- системный администратор;
- ведущий специалист по технической защите информации.

Схема организационной структуры отдела информационной безопасности изображена на рисунке 1.



Рисунок 1 – Схема организационной структуры

Для выполнения функций, задач и реализации прав Отдел взаимодействует со всеми структурными подразделениями ТГУ и сторонними организациями по вопросам осуществления текущей деятельности.

Одной из функций Отдела безопасности является проведение анализа и контроля за состоянием защищенности системы контроля и управления доступом. В настоящее время ТГУ использует систему контроля и управления доступом APACS 3000.

Система контроля и управления доступом (СКУД) — совокупность средств контроля и управления доступом, обладающих технической, информационной, программной и эксплуатационной совместимостью (ГОСТ Р 51241-2008, статья 3.28). Такие системы используются для управления доступом на определённую территорию. Основными элементами такой системы являются: контроллер управления, считыватель и идентификатор.

[11]

Идентификатор хранит код, который служит для определения прав владельца. Обычно идентификатор — это карточка, которую подносят к считывателю. Считыватель получает код идентификатора и передает его в контроллер. Контроллер управления принимает решение о пропуске человека предъявившего идентификатор.

В ТГУ СКУД также может быть использована для сбора статистики о посещаемости студентов, проходивших через точки контроля доступа. Например, можно узнать кто проходил через турникет, в какой день и в какое время.

1.2 Анализ бизнес-процесса аналитики событий СКУД

Рассмотрим уже существующий бизнес-процесс аналитики событий системы контроля и управления доступом (СКУД). Для этого используем методологию реинжиниринга бизнес-процессов [9]. Она состоит из следующих этапов:

- определение причин и целей проведения реинжиниринга бизнес-процесса;
- изучение и анализ существующего бизнес-процесса для понимания его деталей и проблем;
- разработка модели нового бизнес-процесса, который устраняет неэффективности и соответствует целям реинжиниринга;
- внедрения нового бизнес-процесса;
- оценка эффективности нового процесса и внесение корректировок и улучшений при необходимости.

В качестве нотации для моделирования бизнес-процессов была выбрана BPMN (Business Process Modeling Notation), так как это наиболее популярная и широко используемая нотация в современных системах управления бизнес-процессами. BPMN представляет собой метод иллюстрации бизнес-процессов, использующий диаграмму, аналогичную блок-схеме. Этот метод

поддерживается Группой управления объектами (OMG) и обеспечивает стандартный и понятный способ определения и анализа бизнес-процессов в государственном и частном секторе. BPMN имеет стандартную нотацию, которая понятна менеджерам, аналитикам и разработчикам. [12]

Основная цель BPMN состоит в устранении коммуникационных проблем между различными отделами внутри организации. С помощью диаграммы, созданной в BPMN, можно создать набор основных элементов, которые позволяют как техническим, так и нетехническим специалистам легко понять задействованные процессы. Элементы в BPMN разделены на три основные группы, которые называются объектами потока, соединительными объектами и дорожками. [12]

Для разработки BPMN-диаграмм был использован бесплатный инструмент для моделирования бизнес-процессов BPMN.Studio[21].

На рисунке 2 представлена BPMN-диаграмма «КАК ЕСТЬ» бизнес-процесса аналитики событий СКУД. Участниками бизнес-процесса являются пользователь и программный комплекс APACS 3000. Для того, чтобы просмотреть посещаемость студентов пользователю нужно запустить клиентское приложение APACS 3000 с компьютера на операционной системе Windows и ввести логин и пароль. Если логин и пароль верны, тогда пользователь может запросить отчет аналитики событий. APACS 3000 формирует отчет и предоставляет его пользователю.

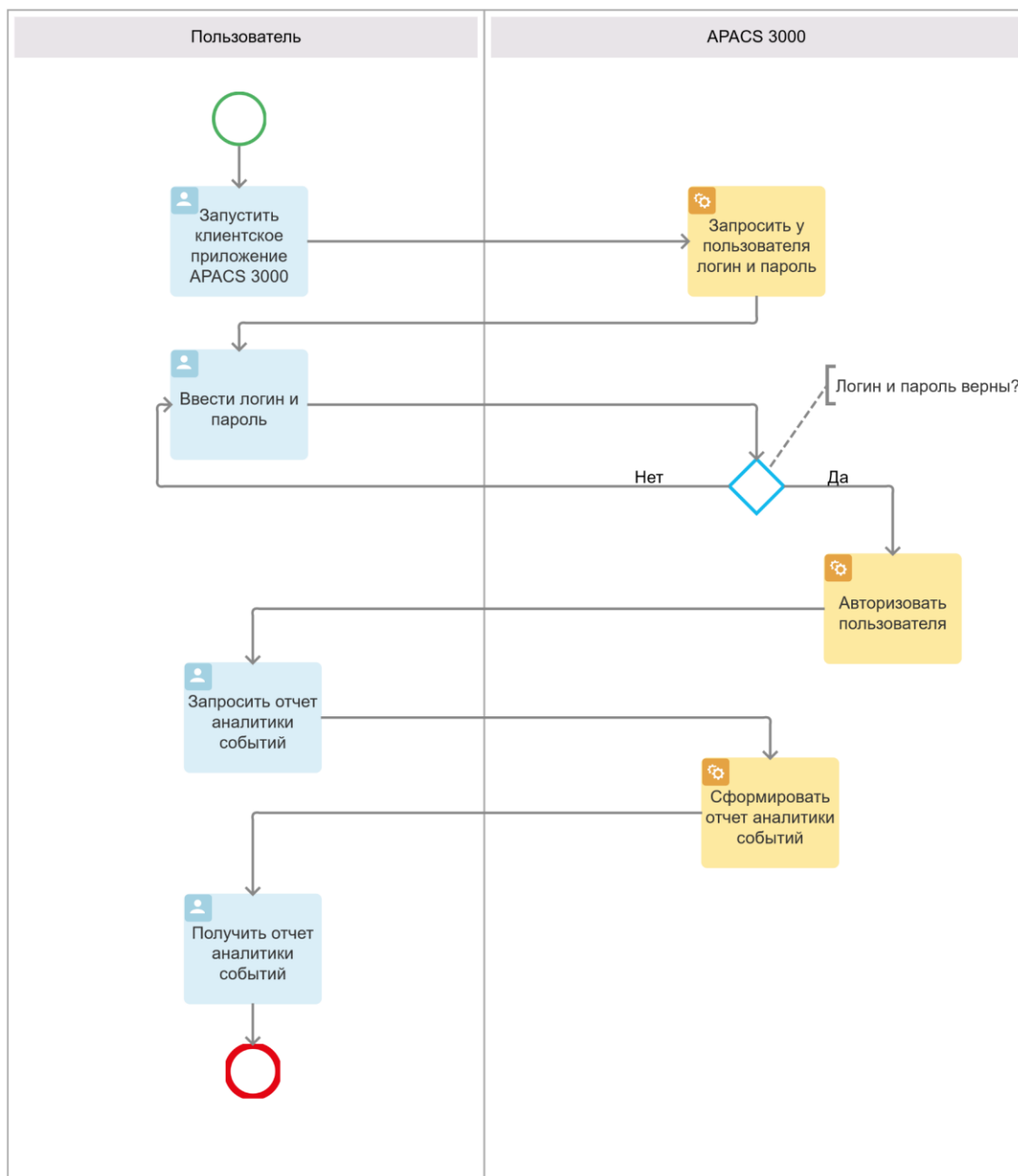


Рисунок 2 – BPMN-диаграмма «КАК ЕСТЬ» бизнес-процесса аналитики событий СКУД

Анализ существующего бизнес-процесса позволил выявить его основной недостаток – с APACS 3000 можно работать только на операционных системах Windows.

В усовершенствованном бизнес-процессе будет три участника: пользователь, веб-приложение для аналитики СКУД и база данных, которую

использует APACS 3000, FireBird. BPMN-диаграмма «КАК ДОЛЖНО БЫТЬ» бизнес-процесса аналитики событий СКУД представлена на рисунке 3.

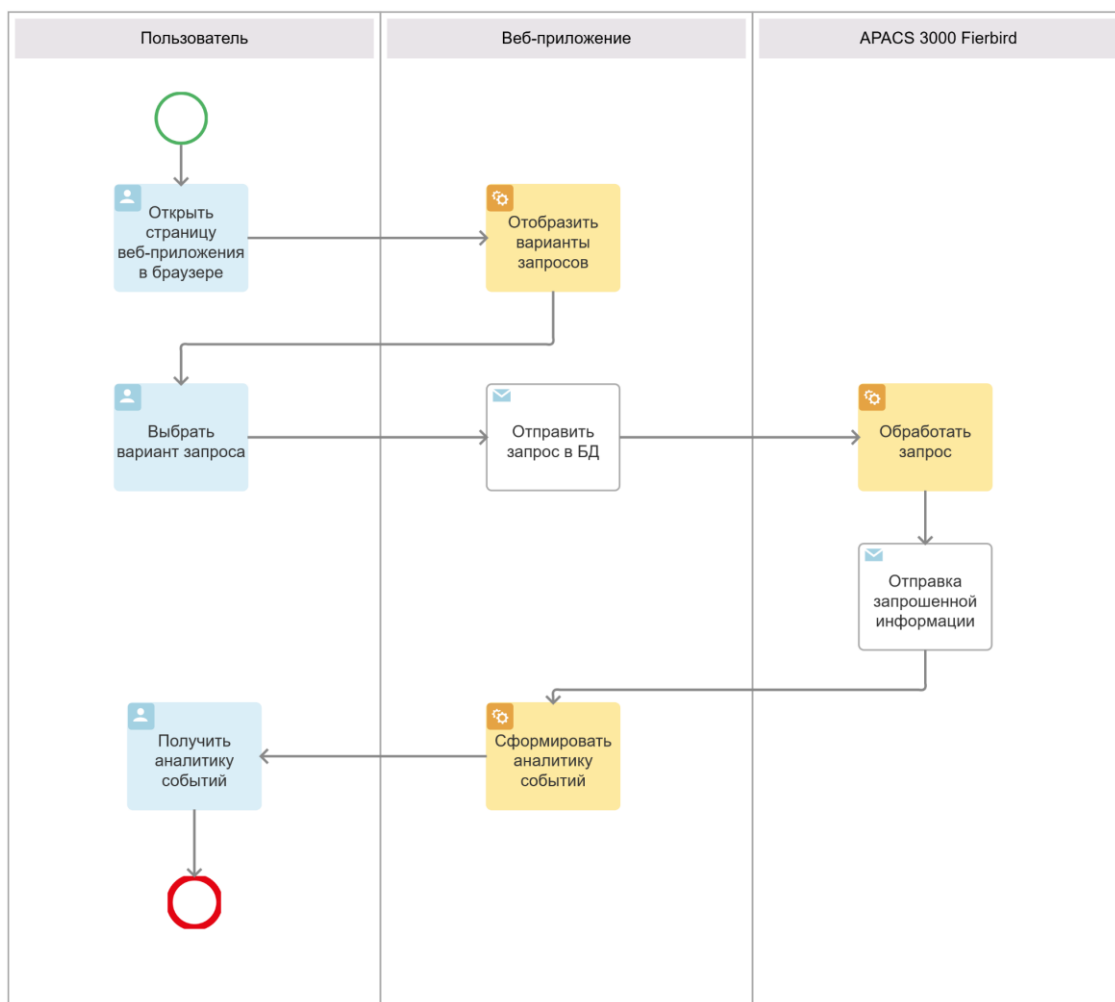


Рисунок 3 – BPMN-диаграмма «КАК ДОЛЖНО БЫТЬ» бизнес-процесса аналитики событий СКУД

Улучшенный бизнес-процесс аналитики событий СКУД состоит из следующих этапов:

- пользователь заходит в браузер и открывает страницу веб-приложения;
- из предложенных веб-приложением вариантов запросов пользователь выбирает нужный ему запрос;
- веб-приложение отправляет запрос в базу данных FireBird;

– когда веб-приложение получает данные, запрошенные от FireBird, он формирует аналитику событий СКУД и отправляет её пользователю.

Как следует из представленной модели использование веб-приложения решает не только проблему просмотра аналитики событий СКУД на разных платформах, но и предоставляет возможность просматривать посещаемость на устройствах без установленного APACS 3000.

1.3 Постановка задачи на разработку веб-приложения для аналитики событий СКУД

Для определения потребностей компании может быть использован простой подход, основанный на модели FURPS+.[17]

FURPS+ – это модель классификации требований к программному обеспечению. Она является расширением модели FURPS, которая означает Функциональность, Удобство использования, Надежность, Производительность и Поддерживаемость. FURPS+ добавляет дополнительные категории, которые следует учитывать при разработке программного обеспечения.

Функциональные требования определяют конкретные задачи, операции или сервисы, которые должны быть доступны в программном обеспечении. Они указывают на функции, которые пользователь ожидает от системы, а также на действия и результаты, которые должны быть достигнуты.

В модели FURPS+ также присутствует раздел «Нефункциональность», который охватывает нефункциональные требования. Нефункциональные требования определяют характеристики и ограничения, связанные с качеством программного обеспечения, а не с его функциональностью. Эти требования определяются с учетом аспектов, таких как производительность, надежность, безопасность, удобство использования и другие.

В таблице 1 представлены основные требования к веб-приложению для аналитики событий СКУД с учетом особенностей методологий FURPS+.

Таблица 1 – Требования к клиентской части веб-приложения

Требование	Статус	Полезность	Риск	Стабильность
Functionality – Функциональные требования				
«Просмотр посещаемости	Одобренное	Критическая	Средний	Низкая
Usability – Требования к удобству использования				
Современный веб-дизайн	Одобренное	Критическая	Средний	Низкая
Отсутствие функциональной избыточности	Одобренное	Критическая	Средний	Низкая
Reliability – Требования к надежности				
Допустимое время простоя – до 60 часов в год, но не более 9 часов в месяц	Одобренное	Важная	Средний	Средняя
Среднее время сбоев: 1 рабочий день	Одобренное	Важная	Средний	Средняя
Возможность восстановления системы после сбоев: 1 рабочий день	Одобренное	Важная	Средний	Средняя
Режим работы: 24x7	Одобренное	Важная	Средний	Средняя
Performance – Требования к производительности				
Допустимое количество одновременно работающих пользователей: 15	Предложенное	Важная	Средний	Средняя
Время реакции на возникновение аварийной ситуации: 10 с	Предложенное	Важная	Средний	Средняя
Supportability – Требования к поддержке»[17]				

Продолжение таблицы 1:

Требование	Статус	Полезность	Риск	Стабильность
«Время устранения критических проблем: в течение рабочего дня	Предложенное	Важная	Средний	Средняя
Проектные ограничения				
Использование тестовой базы APACS 3000	Одобренное	Критическая	Средний	Низкая
Применение современных веб-технологий	Одобренное	Важная	Средний	Низкая
Тип лицензии Freeware	Одобренное	Важная	Средний	Низкая»[17]

Разработанный перечень требований является основой для проектирования веб-приложения.

1.4 Обзор и анализ аналогов веб-приложений для аналитики событий СКУД

Рассмотрим и проанализируем характеристики существующих аналогов веб-приложений на предмет соответствия установленным требованиям. [8]

«EdgeReader Solo ESR40 — это инновационное решение в области контроля и управления доступом, которое позволяет разместить контроллер с интегрированным считывателем и интерфейсом пользователя непосредственно около двери. Оно представляет собой самостоятельную однодверную сетевую систему контроля доступа, которая обеспечивает простое и удобное удаленное управление, и получение отчетов через стандартный Web-интерфейс. Все данные о пользователях, настройки, администрирование и отслеживание событий могут быть легко выполнены через простой и понятный веб-интерфейс, благодаря встроенному сетевому

порту контроллера ESR40, который подключается к компьютеру через RJ-45 коннектор для TCP/IP 10/100 Мбит/с Ethernet» [22].

«Контроллер Honeywell NetAXS 4 представляет собой полнофункциональное решение для системы контроля и управления доступом. Благодаря своему веб-интерфейсу, NetAXS-4 предоставляет возможность дистанционного управления доступом пользователей на охраняемый объект, а также мониторингу состояния системы и созданию отчетов. В отличие от других систем, NetAXS-4 не требует установки специализированного программного обеспечения на отдельный компьютер. Все функции доступны через интуитивно понятный веб-интерфейс, который можно открыть в любом браузере. Кроме того, передаваемые по сети данные защищены с использованием современных криптографических протоколов, что обеспечивает надежную защиту.

Особенности NetAXS:

- система позволяет автоматически обновлять сообщения о событиях и тревогах прямо в окне веб-браузера пользователя;
- предусмотрена возможность экспорта отчетов для последующего анализа или сохранения данных в архив;
- система предоставляет функцию назначения различных уровней доступа владельцам карт для повышения безопасности объекта;
- веб-интерфейс системы поддерживает одновременную работу нескольких пользователей с возможностью выбора различных языков;
- система позволяет расширить количество подключаемых считывателей для контроллера NetAXS-2 на 2 двери. NetAXS-2 (на 2 двери)» [3].

Программный комплекс Nedap AEOS имеет существенное преимущество перед другими системами благодаря тому, что его пользовательский интерфейс реализован в виде Web-приложения. «Это обеспечивает ряд преимуществ, таких как:

- для настройки рабочего места достаточно запустить веб-браузер и получить доступ к серверу в соответствии с установленными правами, без необходимости установки специального программного обеспечения;
- обновление программного обеспечения не представляет сложностей, так как процесс происходит одновременно на всех рабочих местах администраторов системы;
- для работы системы нет необходимости увеличивать количество персонала, так как все важные процессы контролируются и управляются из центра, что особенно важно в случае разнесенных территориально объектов;
- система исключает проблемы рассинхронизации базы данных или некорректных записей в ней;
- требования к аппаратной части компьютеров, на которых будут развернуты рабочие места, значительно снижаются;
- загрузка сети существенно уменьшается, так как по сети передается только результат работы сервера, а не массив данных» [24].

Для быстрого реагирования на тревоги и события доступа в системе AEOS можно использовать графический обработчик тревог (Graphical Alarm Handler), который является Web-приложением и напрямую получает сообщения от контроллеров системы. Это позволяет оперативно реагировать на любые внешние воздействия, даже в том случае, если центральный сервер системы недоступен. Для работы с графическим обработчиком тревог достаточно наличия работающей сети и запущенного на рабочем месте администратора любого устройства с любым Web-браузером. Решение кроссплатформенное и может работать под любой современной операционной системой.

Для выбора ИТ-решения для применения в качестве веб-приложения для аналитики событий СКУД используем таблицу 2.

Таблица 2 – Выбор ИТ-решения для применения в качестве веб-приложения для аналитики событий СКУД

Характеристика/балл (0-3)	HID Edge Solo	Honeywell NetAXS-4	Nedap AEOS
Просмотр посещаемости	3	3	3
Использование тестовой базы APACS 3000	2	2	2
Применение современных веб-технологий	1	2	3
Тип лицензии Freeware	0	0	0
Итого:	6	7	8

После проведения анализа имеющихся аналогов было установлено, что ни один из них не удовлетворяет всем установленным требованиям. Исходя из этого, представляется разумным разработать веб-приложение для аналитики событий системы контроля и управления доступом (СКУД).

Выводы по 1 главе

В первой главе исследуется предметная область и формулируется задача разработки веб-приложения для анализа событий в системе контроля и управления доступом (СКУД). В результате проведенных исследований были получены следующие выводы:

- веб-приложение для аналитики событий СКУД позволит пользователям просматривать информацию о посещаемости студентов с любого устройства, подключенного к интернету;
- сравнительный анализ существующих аналогов показал, что ни один из них не удовлетворяет всем поставленным требованиям.

На основании этих выводов было принято решение разработать веб-приложение для аналитики событий СКУД.

2 Логическое проектирование веб-приложения для аналитики событий СКУД

2.1 Выбор технологии логического моделирования веб-приложения

В процессе логического моделирования информационной системы происходит формализация ее бизнес-процессов, структуры данных, логики их взаимодействия и принципов работы. Это позволяет определить ключевые элементы системы, их связи и взаимодействия, а также выявить потенциальные проблемы и улучшить работу системы до ее физической реализации [14].

В результате логического моделирования создается документация, которая описывает функциональные требования к системе, ее архитектуру и принципы взаимодействия. Эта документация является основой для создания физической модели информационной системы, которая включает в себя выбор технологий, разработку кода и настройку оборудования.

Для построения логической модели веб-интерфейса была выбрана нотация UML. UML (Unified Modeling Language) — это стандартизированная нотация, которая позволяет создавать логические модели информационных систем и других объектов с помощью графических символов. Она является одной из наиболее распространенных и популярных нотаций для логического моделирования информационных систем [4].

Выбор UML для логического моделирования имеет несколько преимуществ:

- стандартизация: UML является стандартом, который широко принят в индустрии, что делает ее удобной и понятной для большинства специалистов в области информационных технологий;
- гибкость: UML позволяет создавать различные типы диаграмм, которые могут быть использованы для моделирования различных аспектов

информационной системы, таких как структура данных, бизнес-процессы и взаимодействие между компонентами;

- визуализация: UML предоставляет широкий спектр графических символов, которые позволяют создавать наглядные и понятные диаграммы;

- поддержка инструментария: существует множество инструментов, которые поддерживают создание UML-диаграмм, что делает их создание и использование более удобным;

- удобство в обучении: UML относительно легко осваивается, что делает ее доступной для большинства разработчиков и специалистов в области информационных технологий.

Для разработки логической модели веб-интерфейса было использовано CASE-средство StarUML.

2.2 Построение логической модели веб-приложения

«Логическая модель представляет собой комплекс UML-диаграмм, образующих ядро языка UML. Было построено три UML-диаграммы: диаграмма вариантов использования, диаграмма классов и диаграмма последовательности» [4].

Диаграммы вариантов использования в UML используются для понимания функциональности системы, идентификации акторов и функциональности, визуализации вариантов использования, определения границ системы, коммуникации с заинтересованными сторонами и как основа для дальнейшего моделирования [14].

Диаграммы вариантов использования строятся из следующих основных элементов:

- акторы (Actors): представляют внешние сущности, такие как пользователи, другие системы или внешние устройства, которые взаимодействуют с системой;

- варианты использования (Use Cases): представляют конкретные сценарии или случаи использования системы. Они описывают, как акторы взаимодействуют с системой для достижения определенных целей;

- отношения (Relationships): используются для связывания акторов и вариантов использования, а также между самими вариантами использования. Некоторые распространенные отношения включают включение (include), расширение (extend), обобщение (generalization) и использование (association).

Диаграмма вариантов использования изображена на рисунке 4. Актером, взаимодействующим с системой, является Пользователь. Варианты использования:

- «Сформировать краткую сводку о посещаемости студента» связан отношением ассоциации с «Пользователем» и отношением включения с «Представить количество посещений за неделю» и «Представить информацию о последнем посещении»;

- «Сформировать отчет о посещаемости студента» связан отношением ассоциации с «Пользователем» и отношением включения с «Представить информацию о последнем посещении», «Сформировать аналитику посещаемости за месяц», «Сформировать аналитику посещаемости по месяцам».

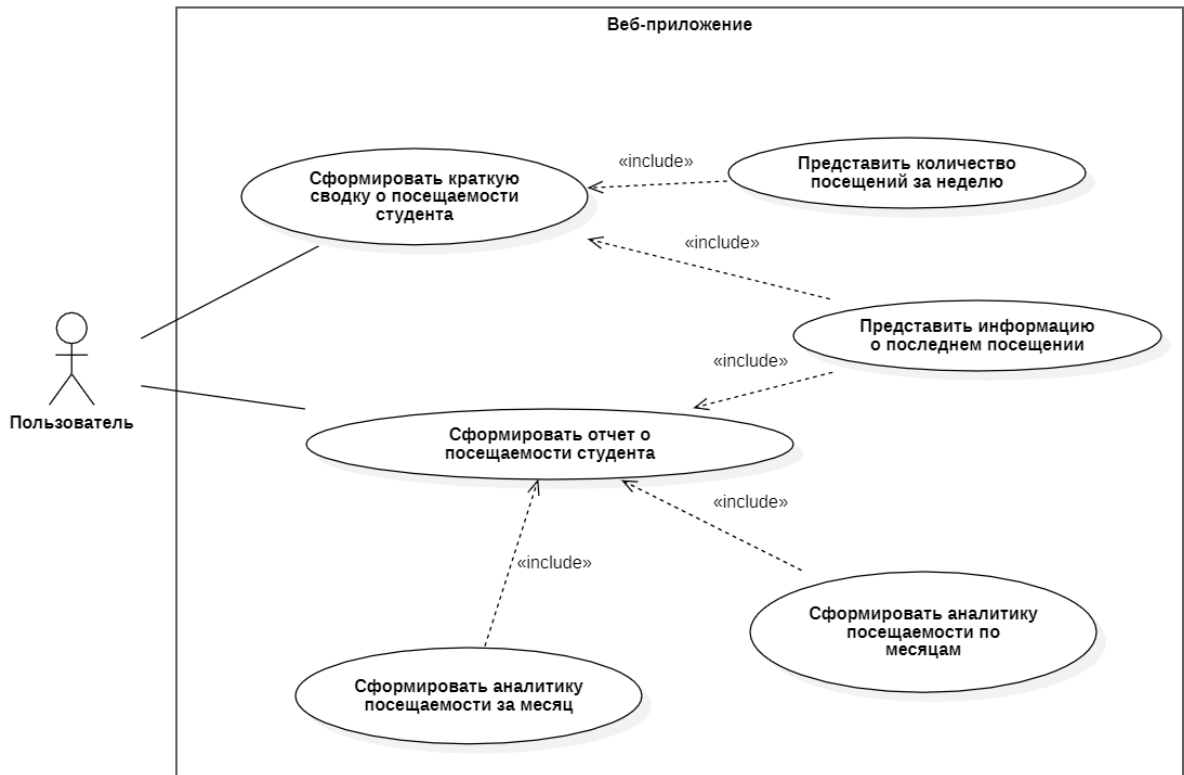


Рисунок 4 – Диаграмма вариантов использования веб-приложения для аналитики СКУД

Диаграмма классов в UML используется для визуализации структуры классов в системе, их атрибутов, методов и взаимосвязей между классами. Она позволяет представить концептуальную модель системы, выделить ключевые классы и их связи, а также описать основные аспекты проектирования объектно-ориентированной системы.

Диаграмма классов моделируемого веб-приложения показана на рисунке 5.

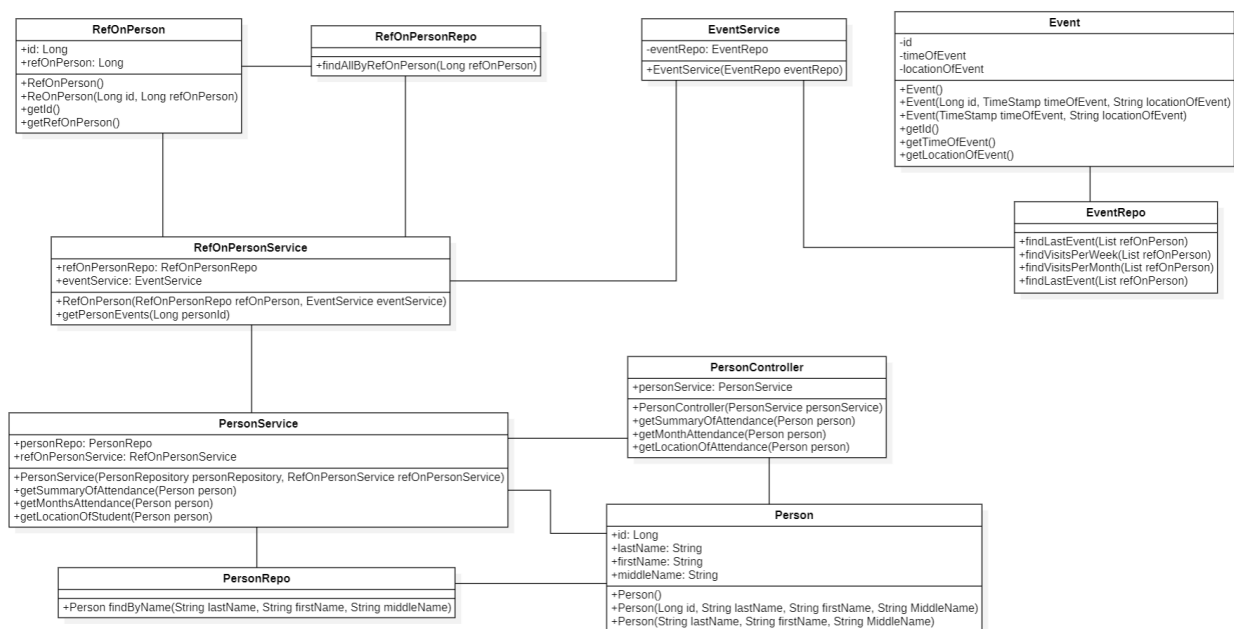


Рисунок 5 – Диаграмма классов моделируемого веб-приложения для аналитики СКУД

В таблице 3 описываются классы веб-приложения для аналитики событий СКУД.

Таблица 3 – Описание классов веб-приложения для аналитики событий СКУД

Класс	Описание
RefOnPerson	Класс, представляющий сущность базы данных
RefOnPersonService	Класс представляющий сервисный слой, работающий с сущностью RefOnPerson
RefOnPersonRepo	Интерфейс для доступа к данным сущности RefOnPerson
PersonService	Класс представляющий сервисный слой, работающий с сущностью Person
PersonRepo	Интерфейс для доступа к данным сущности Person
PersonController	Класс отвечает за обработку входящих запросов REST API

Продолжение таблицы 3:

Класс	Описание
Person	Класс, представляющий сущность базы данных
Event	Класс, представляющий сущность базы данных
EventService	Класс представляющий сервисный слой, работающий с сущностью Event
EventRepo	Интерфейс для доступа к данным сущности Event

Диаграмма последовательности в UML используется для визуализации взаимодействия между объектами или компонентами в рамках определенного сценария или процесса. Она отображает последовательность действий и взаимодействия объектов в процессе выполнения сценария. Кроме того, она используется для уточнения поведения системы путем показа операций, выполняемых на каждом шаге, и передачи сообщений между объектами. Это позволяет получить более глубокое понимание и описать детали поведения системы.

На диаграмме последовательности объекты представлены вертикальными линиями, называемыми линиями жизни. Каждый объект имеет свою линию жизни, на которой показано, когда объект существует в системе. Горизонтальные стрелки, называемые сообщениями, указывают, какие сообщения передаются между объектами.

На рисунке 6 представлена диаграмма последовательности сценария запроса сводки посещаемости студента при помощи моделируемого веб-приложения.

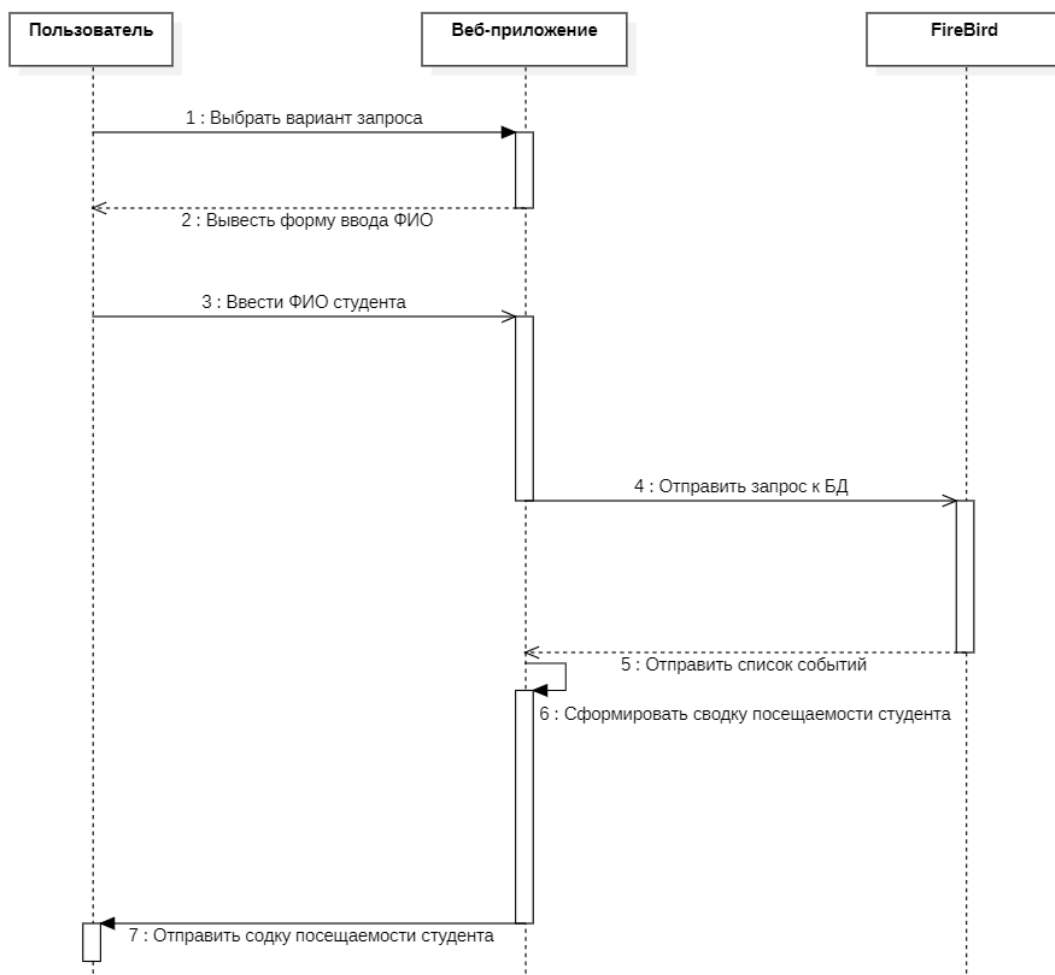


Рисунок 6 – Диаграмма последовательности сценария запроса сводки посещаемости студента

Сценарий запроса сводки посещаемости студента организован следующим образом:

- объект Пользователь передает объекту Веб-приложение выбранный вариант запроса;
- объект Веб-приложение выводит форму ввода ФИО студента для запроса на экран;
- объект Пользователь передает объекту Веб-приложение ФИО студента, для которого нужно сформировать сводку посещаемости;
- объект Веб-приложение запрашивает у объекта FireBird данные, необходимые для формирования сводки посещаемости;

- объект FireBird передает объекту Веб-приложение список запрашиваемых событий;
- объект Веб-приложение формирует сводку посещаемости студента на основе полученного списка событий;
- объект Веб-приложение передает объекту Пользователь сформированную сводку посещаемости студента.

Процесс просмотра сводки посещаемости студента завершен.

2.3 Требования к аппаратно-программному обеспечению веб-приложения

Требования к аппаратно-программному обеспечению APACS 3000, используемого в ТГУ для учета событий СКУД.

«Требования к компьютеру:

- для рабочей станции процессор Intel i3 и выше;
- для сервера процессор Intel i5 и выше;
- оперативная память не менее 4 Гб для рабочей станции и не менее 16 Гб для сервера;
- свободное место на жестком диске не менее 200 Гб;
- видеокарта и монитор с поддержкой разрешения не менее 1024x768 пикселей;
- видеоадаптер захвата изображения, имеющий поддержку Video for Windows (при необходимости);
- последовательный COM порт (RS-232) — количество зависит от используемого оборудования;
- USB–порт.

Требования к программному обеспечению:

- операционная система MS Windows 10/Server 2016/Server 2019 или выше (32 бит для рабочей станции, 64 бит для сервера);
- протокол TCP/IP;

– СУБД Firebird 3.0 (входит в комплект поставки APACS 3000) или Microsoft SQL Server 2019» [11].

Аппаратно-программное обеспечение отдела безопасности ТГУ соответствует требованиям для APACS 3000 и подходит для функционирования веб-приложения для аналитики событий СКУД [8].

Выводы по главе 2

В данной главе проводилось логическое проектирование веб-приложения для аналитики событий СКУД.

На основе результатов проведенной работы были сделаны следующие выводы:

– на стадии логического проектирования используется нотация языка UML;

– для разработки логической модели веб-приложения применяются основные диаграммы UML: диаграмма вариантов использования, диаграмма классов и диаграмма последовательности;

– для корректной работы проектируемого приложения необходимо разработать требования к обеспечению.

Спроектированная логическая модель будет использована для реализации веб-приложения.

3 Реализация и тестирование веб-приложения для аналитики событий СКУД

3.1 Выбор инструментов разработки веб-приложения

Исходными данными для выпускной квалификационной работы является тестовая база данных APACS 3000, которая имеет схожую структуру с базой данных, используемой для актуального учета событий в системах контроля и управления доступом (СКУД) в ТГУ, а также использует систему управления базами данных Firebird.

«APACS 3000 — это комплексное программное обеспечение, предназначенное для управления системами безопасности, контроля доступа и учета рабочего времени. Уникальность APACS 3000 заключается в его способности эффективно функционировать на объектах различных типов и масштабов. Это достигается благодаря его масштабируемости и наличию различных модификаций, позволяющих избегать неоправданного расходования ресурсов на неиспользуемый функционал. База данных APACS 3000 основана на среде Firebird» [11].

«Firebird - это бесплатная система управления реляционными базами данных SQL с открытым исходным кодом, основанная на версии InterBase с открытым исходным кодом, выпущенной Borland Software Corp, ранее известной как Inprise Corp. Разработанный на C и C ++, Firebird поддерживает основные аппаратные и программные платформы, включая Windows, Linux и Mac OS X. Firebird предлагает множество стандартных функций ANSI SQL, позволяет одновременные операции OLTP и OLAP через свою многопоколенную архитектуру и поддерживает хранимые процедуры и триггеры»[15].

Для разработки веб-приложения был выбран язык Java. Java — это объектно-ориентированный язык программирования, созданный компанией Sun Microsystems в 1995 году. Он разработан для создания программ, которые могут выполняться на любой платформе, независимо от аппаратного

обеспечения и операционной системы. Java широко используется для создания приложений, веб-приложений, мобильных приложений и игр. Он также используется в больших корпоративных системах для создания веб-сервисов и приложений, которые могут работать с различными базами данных.

«В качестве фреймворка для разработки веб-приложения был выбран Spring Framework. Spring Framework — это один из самых популярных фреймворков для разработки приложений на языке Java. Он обеспечивает инфраструктуру для разработки приложений с использованием различных компонентов и библиотек, а также упрощает разработку веб-приложений, включая RESTful приложения и приложения на основе сервлетов» [13].

Spring Framework включает в себя множество модулей, каждый из которых предназначен для решения конкретных задач. Например, модуль Spring Core предоставляет инфраструктуру для управления зависимостями и жизненным циклом компонентов, а модуль Spring MVC предоставляет инструменты для создания веб-приложений на основе паттерна Model-View-Controller.

Кроме того, Spring Framework обеспечивает поддержку интеграции с другими технологиями, такими как ORM-фреймворки (например, Hibernate), библиотеки для работы с базами данных (например, JDBC), и т.д.

В целом, Spring Framework предоставляет множество инструментов для упрощения разработки приложений на Java, ускорения процесса разработки и повышения эффективности разработчиков.

3.2. Разработка архитектуры веб-приложения

На стадии физического проектирования приложения используются UML диаграммы пакетов и развертывания.

Диаграмма пакетов – это средство визуализации и организации компонентов и подсистем в программном проекте. Она позволяет логически группировать элементы модели и представляет их в виде иерархической

структуры, отображая связи и зависимости между пакетами. Диаграмма пакетов помогает упростить анализ и понимание проекта, позволяя разбить его на более мелкие и управляемые части.

На рисунке 7 изображена диаграмма пакетов разрабатываемого веб-приложения. На данной диаграмме изображен основной пакет приложения `com.tsu.analytics_of_events`. Он содержит пакеты `event`, `person`, `refOnPerson`. Пакет `com.tsu.analytics_of_events` связан отношениями зависимости с внешними пакетами, так как его внутренние пакеты связаны отношениями зависимости с ними. Внешние пакеты: `org.springframework`, `java.util`, `java.sql`.

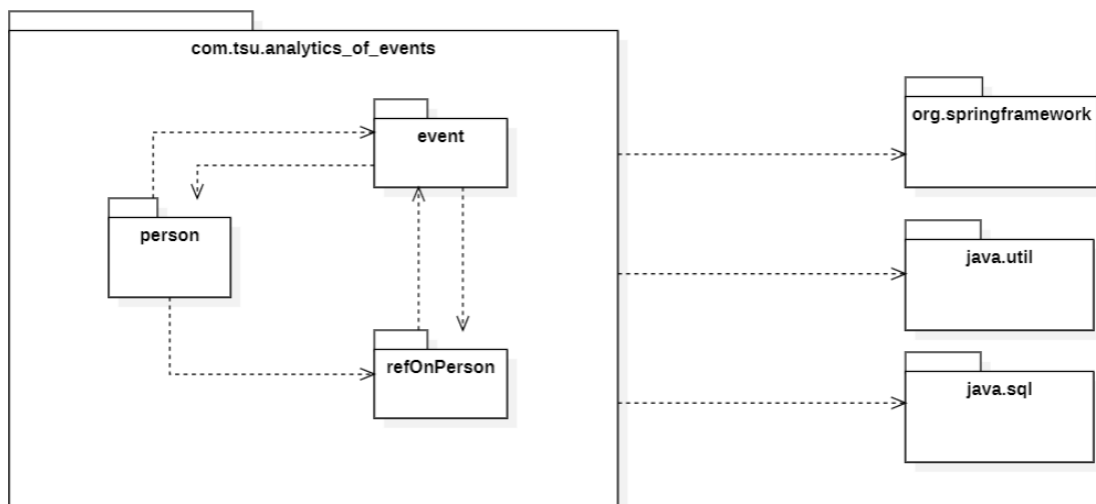


Рисунок 7 – Диаграмма пакетов веб-приложения

«Новая архитектура должна представлять собой трехуровневое приложение и включать в себя клиент (веб-клиент), уровень логики (сервер приложений) и уровень хранения данных (сервер базы данных).

Для создания архитектуры будет применяться шаблон Model-View-Controller (MVC), который позволяет разделить приложение на три компонента: модель, представление и контроллер. Это позволяет модифицировать каждый компонент независимо от других, минимизируя влияние на остальные части приложения» [2].

Модель представляет собой компонент, содержащий данные и методы для работы с ними. Она также отвечает на запросы, изменяя свое состояние при необходимости. Представление отвечает за визуализацию информации и может принимать форму, например, графических элементов или формы. Контроллер служит связующим звеном между пользователем и системой, контролирует ввод данных и использует модель и представление для реализации необходимых реакций. Таким образом, концепция MVC (Model-View-Controller) позволяет разделить данные, их визуализацию и обработку действий пользователя на три отдельных компонента.

Диаграмма развертывания в UML используется для визуального представления физической архитектуры системы и описания распределения компонентов и аппаратного обеспечения в ней.

На рисунке 8 изображена диаграмма развертывания разрабатываемого веб-приложения.

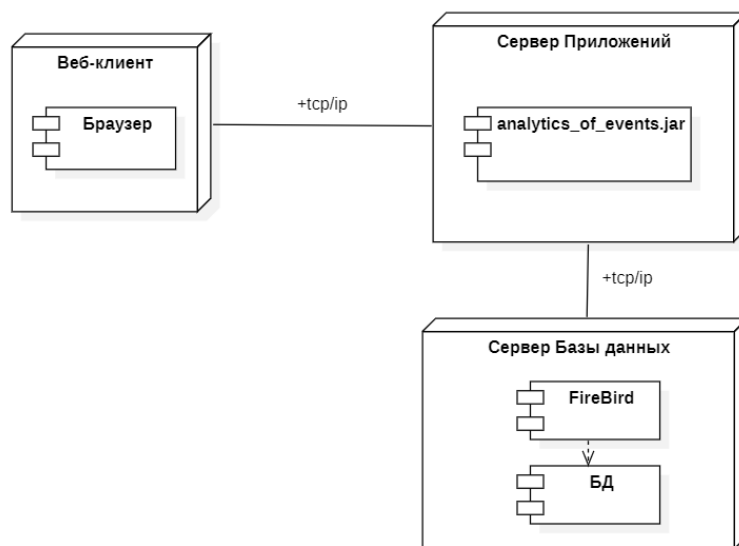


Рисунок 8 – Диаграмма развертывания веб-приложения

На диаграмме изображены три узла: Веб-клиент, Сервер приложений и Сервер базы данных. На Веб-клиенте размещен компонент Браузер, который

взаимодействует с компонентом Веб-приложение, размещенном на Сервере приложений. Веб-приложение взаимодействует с СУБД FireBird, размещенной на Сервере базы данных.

3.3 Разработка физической модели данных веб-приложения

Для разработки физической модели веб-приложения для аналитики событий СКУД была выбрана технология, основанная на методологии IDEF1X.

В рамках методологии IDEF1X для описания данных используются следующие элементы: сущности (entities), атрибуты (attributes), отношения (relationships) и домены (domains). Сущности представляют объекты, для которых в базе данных будет храниться информация, а атрибуты определяют характеристики сущностей. Отношения определяют связи между сущностями, а домены позволяют определить допустимые значения атрибутов.

Для создания физической модели данных была выбрана среда MySQL Workbench, которая поддерживает стандарт IDEF1X. MySQL Workbench представляет собой инструмент для визуального проектирования баз данных, объединяющий функции проектирования, моделирования, создания и управления базами данных в одной интегрированной среде для системы управления базами данных MySQL.

На рисунке 9 изображена физическая модель данных веб-приложения для аналитики СКУД. Из тестовой базы APACS_3000_Demo были взяты только три таблицы, так как именно они необходимы для работы веб-интерфейса.

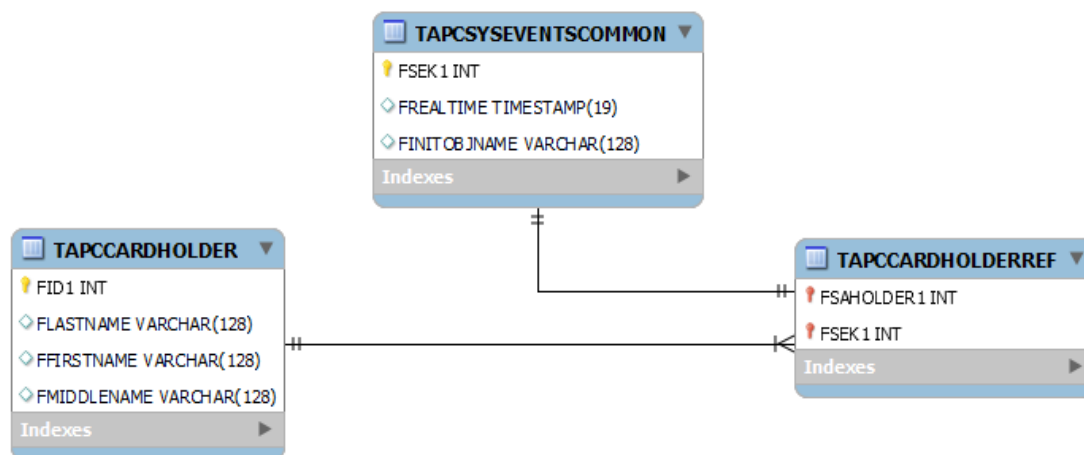


Рисунок 9 – Физическая модель данных веб-приложения для аналитики
СКУД

Таблица TAPCSYSEVENTSCOMMON содержит в себе номер события FSEK1, который является первичным ключем. Поле FREALTIME обозначает время события, FINITOBJNAME содержит местонахождение турникета, считывающего события входа-выхода.

В таблице TAPCCARDHOLDER содержится информация о владельце пропуска: FID1 – идентификатор владельца, FLASTNAME – фамилия, FFIRSTNAME – имя, FMIDDLENAME. – отчество.

TAPCCARDHOLDERREF содержит поля FSEK1 и FSAHOLDER1, которые являются идентификаторами таблиц TAPCSYSEVENTSCOMMON и TAPCCARDHOLDER.

3.4 Разработка веб-приложения для аналитики событий СКУД

Для разработки веб-приложения необходимо создать Spring приложение и подключить необходимые зависимости. Так как для сборки проекта используется maven, зависимости подключаются в pom.xml. Для работы с FireBird нужно подключить JDBC драйвер, который можно найти на официальном сайте FireBird. Также подключаются зависимости:

- для работы с Spring Web и MVC `spring-boot-starter-web`;
- для написания тестов `spring-boot-starter-test`;
- шаблонизатор `spring-boot-starter-freemarker` для генерации текстового вывода.

Далее создается файловая структура приложения. Для каждой используемой таблицы базы данных создается Entity. Она является объектным представлением таблицы и позволяет работать таблицей как с объектами. Для этого необходимо описать все поля таблицы как атрибуты класса и добавить геттеры, сеттеры, конструкторы класса.

На рисунке 10 изображено представление таблицы TAPCCARDHOLDER в виде сущности Person. Каждое поле таблицы соответствует атрибуту класса Person. Также для каждого атрибута написаны геттеры и сеттеры.

```
@Entity
@Table(name = "TAPCCARDHOLDER")
public class Person {
    2 usages
    @Id
    @Column(name = "FID1")
    private Long id;
    3 usages
    @Column(name = "FLASTNAME")
    private String lastName;
    3 usages
    @Column(name = "FFIRSTNAME")
    private String firstName;
    3 usages
    @Column(name = "FMIDDLENAME")
    private String middleName;
```

Рисунок 10 – Код файла Person

Для доступа к данным создаются репозитории для каждой используемой таблицы в базе данных. Репозиторий использует Spring Data JPA, который позволяет генерировать запросы к базе данных из имён методов или же используя аннотацию `@Query`. На рисунке 11 показан репозиторий `PersonRepository`, в котором содержится метод для поиска человека по фамилии, имени и отчеству.

```
package com.tsu.analytics_of_events_of_the_access_control.person;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.Optional;

2 usages  🚩 DdcoldD
@Repository
public interface PersonRepository extends JpaRepository<Person, Long> {
    2 usages  🚩 DdcoldD
    Optional<Person> findByLastNameAndFirstNameAndMiddleName(String lastName, String firstName, String middleName);
}
|
```

Рисунок 11 – Код файла `PersonRepository`

В сервисной части приложения находится основная логика приложения, она обозначается аннотацией `@Service`. На рисунке 12 изображен фрагмент кода класса `PersonService`.

```

@Service
public class PersonService {
    3 usages
    private final PersonRepository personRepository;
    2 usages
    private final RefOnPersonService refOnPersonService;
    no usages  DdcoldD
    @Autowired
    1 public PersonService(PersonRepository personRepository, RefOnPersonService refOnPersonService) {
        this.personRepository = personRepository;
        this.refOnPersonService = refOnPersonService;
    1 }
    no usages  new *
    1 public Person getPerson(Person person) throws Exception{
        String lastName = person.getLastName();
        String firstName = person.getFirstName();
        String middleName = person.getMiddleName();
        Optional<Person> person1 = personRepository.findByLastNameAndFirstNameAndMiddleName(lastName,firstName,middleName);
    1 if(person1.isPresent()){
        return person1.get();
    1 }else{
        throw new Exception("Person doesn't exists");
    1 }
    1 }
}

```

Рисунок 12 – Код файла PersonService

Контроллер обрабатывает запрос пользователя и возвращает результат, обозначается аннотацией `@Controller`. На рисунке 13 изображен фрагмент класса `PersonController`. В аннотации `@RequestMapping` содержится путь для запроса к контроллеру. Функция `getPersonByName` обрабатывает запрос по ФИО человека и возвращает список событий на клиента.

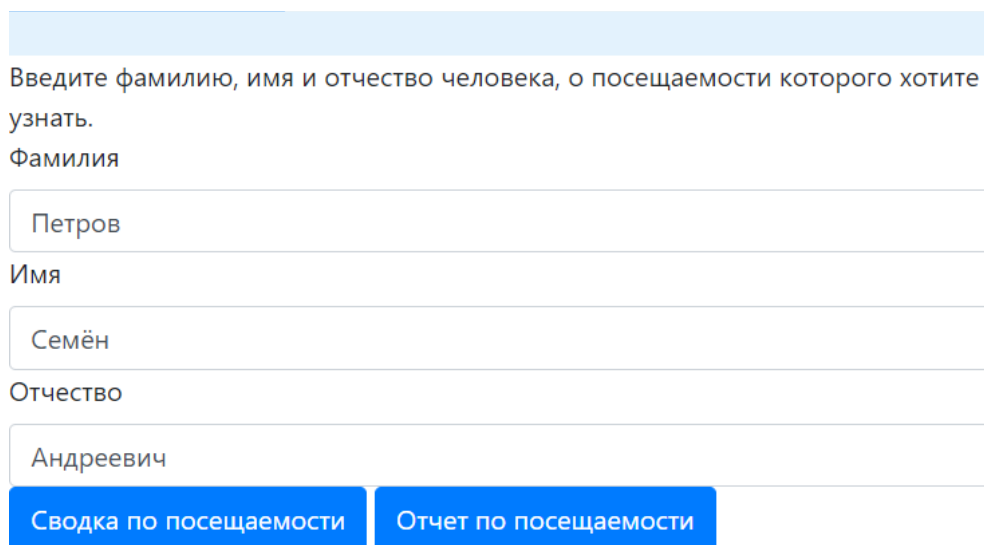
```

@RestController
@RequestMapping(path = "/api/v1/person")
public class PersonController {
    2 usages
    private final PersonService personService;
    no usages  DdcoldD
    @Autowired
    public PersonController(PersonService personService) { this.personService = personService; }
    no usages  DdcoldD
    @GetMapping(path = "/")
    public List<Event> getPersonByName(@RequestBody Person person) throws Exception {
        return personService.getPersonEvents(person);
    }
}

```

Рисунок 13 – Код файла PersonController

Для того, чтобы пользователь мог взаимодействовать с веб-приложением, используются HTML-страницы. На рисунке 14 представлен скриншот веб-страницы index.html.



Введите фамилию, имя и отчество человека, о посещаемости которого хотите узнать.

Фамилия

Имя

Отчество

Рисунок 14 – Скриншот веб-страницы index.html

Если ввести в форму ФИО и выбрать, например «Отчет по посещаемости», то выведется отчет по посещаемости. На рисунке 15 изображен пример отчета.

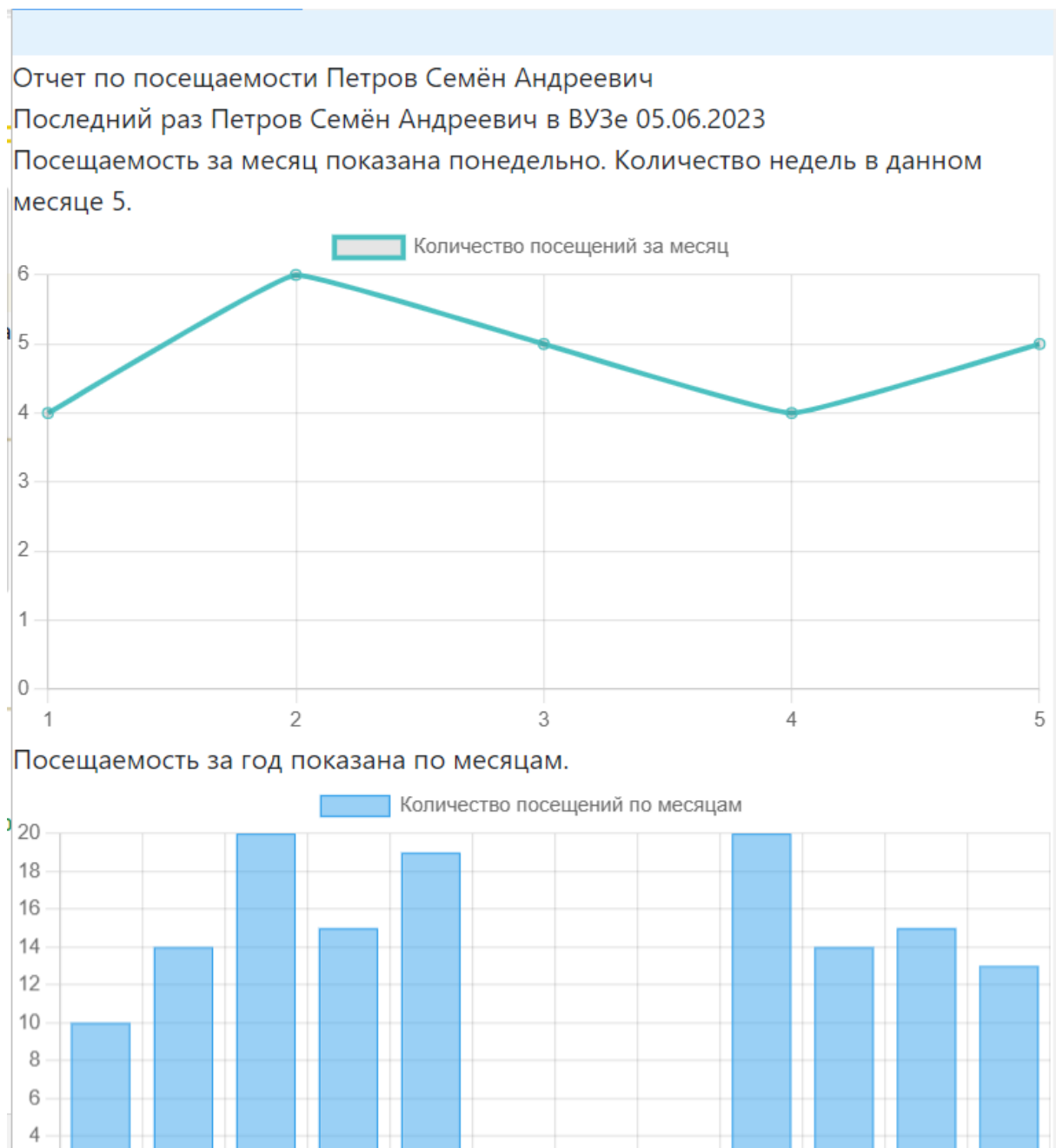


Рисунок 15 – Скриншот отчета по посещаемости

В отчете есть информация о последнем посещении, посещаемости за неделю в виде линейной диаграммы и посещаемости за год в виде гистограммы.

3.5 Тестирование веб-приложения

В рамках дипломной работы, важным этапом является тестирование Spring-приложения. Это позволит проверить правильность работы приложения и удостовериться в его функциональности.

Модульные тесты позволяют проверить работу отдельных компонентов приложения, включая сервисы, репозитории и контроллеры. Для этого рекомендуется использовать соответствующий фреймворк для модульного тестирования, например, JUnit или TestNG. Создание тестовых случаев для каждого компонента поможет удостовериться в их правильном функционировании. Примеры таких тестов включают проверку возвращаемых значений методов, обработку исключений и взаимодействие с базой данных.

На рисунке 16 изображен фрагмент кода интеграционного теста для веб-приложения.

```

@SpringBootTest
public class AnalyticsOfEventsOfTheAccessControlApplicationTests {

    1 usage
    @Mock
    private PersonRepository personRepository;

    1 usage
    @InjectMocks
    private PersonService personService;

    no usages new *
    @BeforeEach
    public void setup() {
        MockitoAnnotations.openMocks( testClass: this);
    }

    no usages DdcoldD *
    @Test
    public void testSomeMethod() {
        // Установка предусловий
        when(personRepository.getData()).thenReturn( t "Test Data");

        // Выполнение метода, который нужно протестировать
        String result = personService.someMethod();

        // Проверка результата
        assertEquals( expected: "Processed Test Data", result);
    }
}

```

Рисунок 16 – Фрагмент кода интеграционного теста

AnalyticsOfEventsOfTheAccessControlApplicationTests является классом модульных тестов, предназначенным для тестирования функциональности сервисного класса PersonService в приложении управления доступом к аналитике событий. В процессе тестирования, взаимодействие PersonService с репозиторием PersonRepository моделируется с помощью заглушек (mock) зависимостей, которые инъецируются в PersonService с помощью аннотаций @Mock и @InjectMocks.

Перед каждым запуском тестов, аннотация @BeforeEach обеспечивает настройку тестовых случаев. Внутри метода testSomeMethod мы устанавливаем предусловия для заглушки PersonRepository.getData(), указывая, что она должна возвращать "Test Data". Затем мы вызываем метод

someMethod объекта PersonService и сохраняем результат в переменную result. Наконец, мы проверяем, что фактический результат соответствует ожидаемому результату "Processed Test Data" с использованием метода assertEquals().

Результат тестирования изображен на рисунке 17.

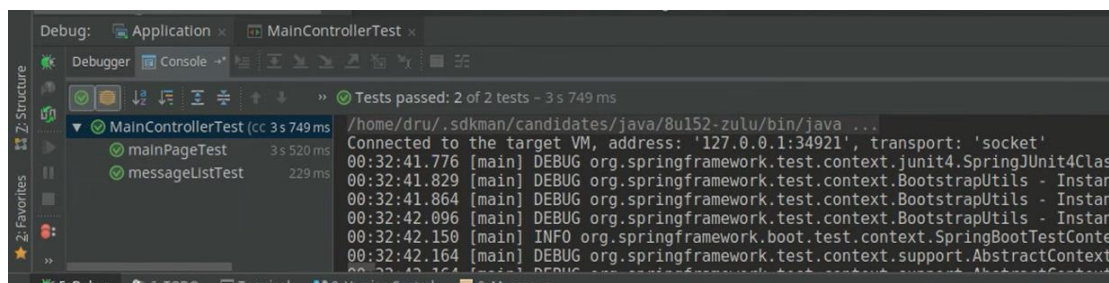


Рисунок 17 – Результаты пройденных тестов

Все тесты пройдены, что означает корректную работу программы.

Выводы по главе 3

Третья глава посвящена реализации веб-приложения для аналитики событий СКУД.

В данной главе были выбраны средства разработки веб-приложения. Затем была построена физическая модель разрабатываемого приложения и продемонстрирована физическая модель базы данных.

Было реализовано спроектированное веб-приложение для аналитики событий СКУД, продемонстрирована работа веб-приложения. Также были разработаны интеграционные тесты.

Заключение

Итогом выпускной квалификационной работы являлась разработка веб-приложения для аналитики событий СКУД.

Была проанализирована деятельность Отдела безопасности ТГУ, выявлено, что одной из функций Отдела безопасности является проведение анализа и контроля за состоянием защищенности системы контроля и управления доступом. Был проанализирован существующий бизнес-процесс и выявлен его недостаток – отсутствие кроссплатформенности существующей системы для аналитики событий СКУД. Были проанализированы существующие ИТ-решения и поставлена задача на разработку веб-приложения для аналитики событий СКУД.

Была построена логическая модель разрабатываемого веб-приложения в виде комплекса UML диаграмм. Также были разработаны аппаратные и программные требования для корректной работы веб-приложения.

Были выбраны инструменты разработки: язык разработки Java, фреймворк Spring, язык гипертекстовой разметки HTML. В качестве базы данных была взята APACS 3000, и реляционная система управления базами данных FireBird. Была разработана архитектура веб-приложения: диаграмма пакетов и диаграмма развертывания. Была представлена физическая модель базы данных.

Была описана разработка веб-приложения и демонстрация его работы. Также было проведено интеграционное тестирование.

Моя выпускная бакалаврская работа является актуальной, так как предлагает использование локального веб-приложения, которое не требует установки на каждый компьютер в Отделе безопасности ТГУ или в другой организации, работающей с APACS 3000.

Список используемой литературы

1. Васильев, А.Н. Java. Объектно-ориентированное программирование: Учебное пособие: для магистров и бакалавров. Базовый курс по объектно-ориентированному программированию / А.Н. Васильев. - СПб.: Питер, 2013. - 400 с.
2. Динеш Р. Spring. Все паттерны проектирования - Питер, 2019 г. – 320 с., ISBN: 978-5-4461-0935-7.
3. Киселев С. В. Веб-дизайн / С. В. Киселев, С. В. Алексахин, А. В. Остроух – Москва, Академия, 2019 г.- 64 с.
4. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с.
5. МакГрат, М. Программирование на Java для начинающих / М. МакГрат. - М.: Эксмо, 2016. - 192 с.
6. Машнин, Т.С. Web-сервисы Java. Профессиональное программирование / Т.С. Машнин. - СПб.: BHV, 2012. - 560 с.
7. Минник Крис, Титтел Эд. HTML5 и CSS3 для чайников. — М.: Диалектика, 2019. — 400 с.
8. Мкртычев С.В., Гущина О.М., Очеповский А.В. Прикладная информатика. Бакалаврская работа [Электронный ресурс]: электрон. учеб.-метод. пособие. Тольятти. ТГУ: Изд-во ТГУ, 2019. 1 оптический диск.
9. Молоткова Н. В., Хазанова Д. Л. Реинжиниринг бизнес-процессов [Электронный ресурс]: учебное пособие. Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2019. 81 с.
10. ПОЛОЖЕНИЕ Об отделе информационной безопасности, МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное бюджетное образовательное

учреждение высшего образования «Тольяттинский государственный университет», приложение 3 к приказу №1353 от 11.08.2022.

11. Программный комплекс APACS 3000 [Электронный ресурс] — Заглавие с экрана. Режим доступа: https://www.aamsystems.ru/programmnye_kompleksy/programmnyu_kompleks_apacs_3000/ (дата обращения: 25.03.2023).

12. Репин В. В. Моделирование бизнес-процессов в нотации BPMN. Пособие для начинающих. Часть I. — Издательские решения, 2019. — 84 с. — ISBN 978-5-44966989-6.

13. Уоллс К. Spring в действии, ДМК-Пресс, 2022 г., ISBN: 978-5-93700-112-2.

14. Фаулер М. UML. Основы, 3-е издание. — Пер. с англ. — СПб: СимволПлюс, 2004. — 192 с., ил. ISBN 5-93286-060-X.

15. Симонов Д. Руководство по языку SQL СУБД Firebird 2.5: Firebird 2.5.9. v.0500-1 / Д. Симонов, П. Винкенуг, Д. Филиппов, Д. Еманов, А. Карпейкин, Д. Кузьменко, А. Ковязин — Москва: Московская биржа, IBSurgeon, 2020. — 576 с.

16. Firebirdsql [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://firebirdsql.org/en/reference-manuals/> (дата обращения: 08.04.2023).

17. What is the use of FURPS+ model in classifying requirements? [Электронный ресурс]. URL: <https://findanyanswer.com/what-is-the-use-of-furps-model-in-classifying-requirements>

18. Bloch, Joshua. "Effective Java." Addison-Wesley Professional, 2017.

19. Spring guides [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://spring.io/guides> (дата обращения: 07.04.2023).

20. Sharma, B.M. "Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools." Apress, 2020.

21. BPMN.Studio [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://bpmn.studio/ru> (дата обращения: 10.03.2023).

22. HID EDGE Solo [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://www.hidglobal.com/products/edge-solo> (дата обращения: 19.03.2023).

23. NetAXS-4 [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://www.security.honeywell.com/ee/product-repository/netaxs-4> (дата обращения: 19.03.2023).

24. Nedar AEOS [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://www.nedapsecurity.com/solutions/aeos-access-control/> (дата обращения: 19.03.2023).