

Universitatea de Stat din Moldova
Facultatea Matematică și Informatică
Departamentul Informaticii Aplicate

Teză de licență

***Spring Security:
Implementarea aplicațiilor
sigure în Java***

A efectuat:

studentul grupei IA2303

Gutu Nicolae

Conducător științific:

Lector al Dep. de
Informatică

Ciornei Oleg

Chișinău, 2025

1. Введение

Производственная практика является важной частью образовательного процесса, позволяя студентам применить полученные в вузе знания в реальных условиях и познакомиться с особенностями профессиональной деятельности. Практика была пройдена в ИТ-компании, специализирующейся на разработке и сопровождении программных решений для трейдинга и инвестиций.

Цель практики заключалась в приобретении практических навыков в области информационных технологий, ознакомлении с основами системного администрирования и сопровождения корпоративной ИТ-инфраструктуры. В рамках практики я выполнял задачи под руководством системного администратора, включая обслуживание оборудования, участие в решении инцидентов, настройку программного обеспечения и тестирование внутренних сервисов.

Отчет сопровождается скриншотами, однако в силу невозможности распространения внутренней информации компании, мною предоставлены обобщенные либо абстрактные изображения.

2. Характеристика организации практики

Компания, в которой проходила практика, занимается созданием и технической поддержкой программных продуктов, используемых в сфере финансов, трейдинга и инвестиций. Основной задачей ИТ-отдела является обеспечение бесперебойной работы внутренних и внешних сервисов, поддержка инфраструктуры, безопасность данных, а также техническая поддержка сотрудников.



В отделе системного администрирования используются современные технологии мониторинга, системы обработки заявок, облачные решения, а также внутренние сервисы, такие как CRM и API. ИТ-отдел обслуживает как серверную инфраструктуру, так и пользовательские рабочие станции, обеспечивает развертывание и настройку программного обеспечения, участвует в развитии внутренних сервисов.

В рамках практики я взаимодействовал с тикет-системой, участвовал в диагностике и устранении инцидентов, анализировал логи, настраивал рабочие станции, а также изучал инструменты мониторинга и участвовал в тестировании API. Практика позволила получить целостное представление о работе службы поддержки и основных аспектах сопровождения ИТ-инфраструктуры.

3. Прodelанная работа и использованные технологии

3.1 Ознакомление с ИТ-инфраструктурой и процессами

Практика началась с изучения общей ИТ-структуры компании, основных технических процессов и используемых инструментов. Под руководством системного администратора я ознакомился с архитектурой корпоративной сети, включая рабочие станции сотрудников, серверные мощности и ключевые внутренние сервисы.

Особое внимание было уделено **тикет-системе**, с помощью которой сотрудники направляют запросы в ИТ-отдел. Я изучил типы заявок, порядок их обработки и приоритезации. В процессе работы участвовал в решении нескольких простых запросов, касающихся неисправностей ПК и программного обеспечения. Это помогло мне понять, как выстраиваются внутренние процессы поддержки пользователей и как ИТ-отдел отслеживает и устраняет возникающие инциденты.

3.2 Работа с пользовательским оборудованием

В рамках практики я активно участвовал в обслуживании и настройке пользовательских компьютеров. Это включало:

- 1) **Обработку тикетов**, связанных с неисправностью ПК, включая диагностику проблем с запуском ОС, периферийными устройствами и базовыми настройками системы.
- 2) **Подготовку новых рабочих мест**: установка операционной системы, антивируса, офисного пакета и другого программного обеспечения, настройка учётных записей и корпоративной почты.
- 3) **Плановую замену старых машин** — я участвовал в замене 12 устаревших ПК на новые. Работы включали физическую установку оборудования, подключение к сети, восстановление необходимых данных и проверку корректной работы сервисов.

Выполнение этих задач позволило мне на практике освоить процессы базового ИТ-обслуживания, научиться работать с типовыми корпоративными настройками и лучше понять структуру рабочего ИТ-окружения в компании.

3.3 Работа с серверным оборудованием

Часть задач во время практики касалась серверного оборудования, на котором работают ключевые сервисы компании. Мне была предоставлена возможность ознакомиться с базовыми аспектами сопровождения серверов, а также с инструментами для диагностики их состояния.

Под руководством системного администратора я изучал работу с RAID-массивами, используя утилиту mdadm, применяемую в системах на базе Linux для управления программным RAID. Я ознакомился с основными командами для просмотра состояния массивов, определением ошибок, проверкой синхронизации и текущего состояния дисков.

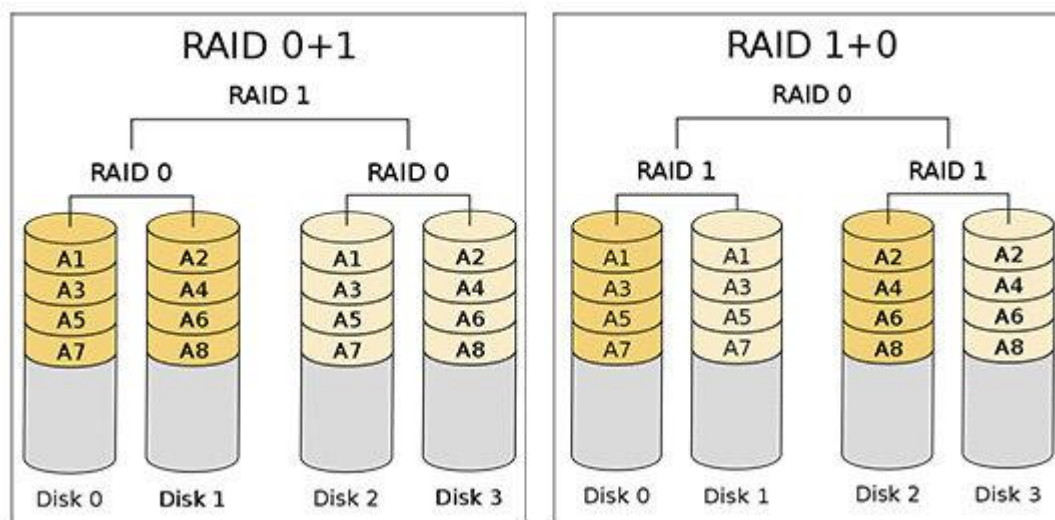


Рис.1 Структура RAID-массивов

Также я проводил проверку состояния SSD и HDD, установленных на сервере. Анализ производился с использованием базовых диагностических утилит (mdadm & smartctl) и логов. По результатам мной был подготовлен простой отчет о текущем техническом состоянии дисков, с указанием времени наработки, температуры, состояния S.M.A.R.T. и возможных признаков износа.

```

Every 2.0s: mdadm --detail /dev/md2

/dev/md2:
  Version : 1.2
  Creation Time : Thu Aug 23 14:12:44 2018
  Raid Level : raid1
  Array Size : 5238784 (5.00 GiB 5.36 GB)
  Used Dev Size : 5238784 (5.00 GiB 5.36 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Thu Aug 23 14:19:16 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

    Name : adair1:2 (local to host adair1)
    UUID : 74780974:f77c292a:914f35bf:458ed15b
    Events : 20

   Number  Major   Minor   RaidDevice State
    -----
     0         8       81         0   active sync  /dev/sdf1
     1         8       97         1   active sync  /dev/sdg1

```

Рис.2 Работа с утилитой «mdadm»

Эти задачи позволили на практике изучить основы мониторинга "железа" серверов и получить представление о процедурах, необходимых для поддержания стабильности инфраструктуры.

3.4 Использование системы мониторинга Grafana

Для наблюдения за состоянием серверов и внутренних сервисов в компании используется система мониторинга **Grafana**. Этот инструмент позволяет в реальном времени визуализировать показатели, собранные с различных источников: нагрузки на процессор, потребление памяти, сетевой трафик, доступность сервисов и прочее.



В рамках практики я ознакомился с принципами работы **Grafana**:

- 1) Узнал, как подключаются источники данных (например, Prometheus, InfluxDB).
- 2) Как создаются дашборды для визуализации метрик.
- 3) Как настраиваются (уведомления в случае выхода показателей за пределы нормы).



Рис.3 Графический интерфейс «Grafana»

Также я изучил существующие в компании дашборды, используемые для отслеживания работы ключевых компонентов инфраструктуры, включая мониторинг времени отклика баз данных и API. Изучение **Grafana** позволило мне понять, как осуществляется мониторинг и раннее обнаружение проблем в корпоративной среде.

3.5 Анализ логов и устранение инцидентов

В процессе практики мне довелось участвовать в анализе инцидента, связанного с недоступностью сайта компании. Пользователи начали получать ошибку **504 Gateway Timeout**, и я ассистировал при диагностике причины сбоя. Проблема оказалась на стороне backend-приложения — сервис на Node.js не дожидался ответа от базы данных, из-за чего зависал.



Рис.4 Node.js Logo

Под руководством администратора мы проверили состояние серверов, службы базы данных и связующие компоненты. После перезапуска ряда сервисов сайт возобновил работу, однако для предотвращения повторных случаев потребовался глубокий анализ логов.

Я изучал логи приложения Node.js, в которых фиксировались все входящие запросы. Чтобы упростить поиск проблемных участков, я написал небольшой **Python**-скрипт, фильтрующий записи по задержкам ответа и повторяющимся IP.

```
log_pattern = re.compile(
    """
    \[(?P<datetime>.*?)\] (?P<level>INFO|ERROR) -
    (?P<ip>\d{1,3}(?:\.\d{1,3}){3}) -
    (?P<method>GET|POST|PUT|DELETE) (?P<path>/\S+) -
    (?P<status>\d{3}) -
    (?P<duration>\d+)ms'
    """
)

slow_requests = []
with open(LOG_FILE, 'r', encoding='utf-8') as f:
    for line in f:
        match = log_pattern.search(line)
        if match:
            duration = int(match.group('duration'))
            if duration > SLOW_RESPONSE_THRESHOLD:
                slow_requests.append({
                    'timestamp': match.group('datetime'),
                    'ip': match.group('ip'),
                    'method': match.group('method'),
                    'path': match.group('path'),
                    'duration_ms': duration
                })

print(f'Количество медленных запросов: {len(slow_requests)}\n')
for req in slow_requests:
    print(f"[{req['timestamp']}] {req['ip']} {req['method']} {req['path']} - {req['duration_ms']}ms")
```

Рис.5 Python скрипт для фильтрации логов

Это позволило выявить один эндпоинт, который отправлял чрезмерное количество обращений к базе данных. Выяснилось, что после недавнего обновления из-за логической ошибки в коде происходила рекурсивная выборка данных.

После локализации причины мной был подготовлен технический отчёт, в котором я описал ход анализа, выявленные узкие места и предложенные временные меры. Этот эпизод стал для меня ценным практическим опытом диагностики инцидентов в живой инфраструктуре.

3.6 Работа с CRM-системой и производительностью БД

Во второй половине практики мне довелось столкнуться с проблемой замедления корпоративной CRM-системы. Пользователи начали жаловаться на долгие отклики, особенно при поиске и фильтрации информации. Совместно с сисадмином мы начали с анализа логов и мониторинга состояния сервисов.

CRM – CUSTOMER RELATIONSHIP MANAGEMENT

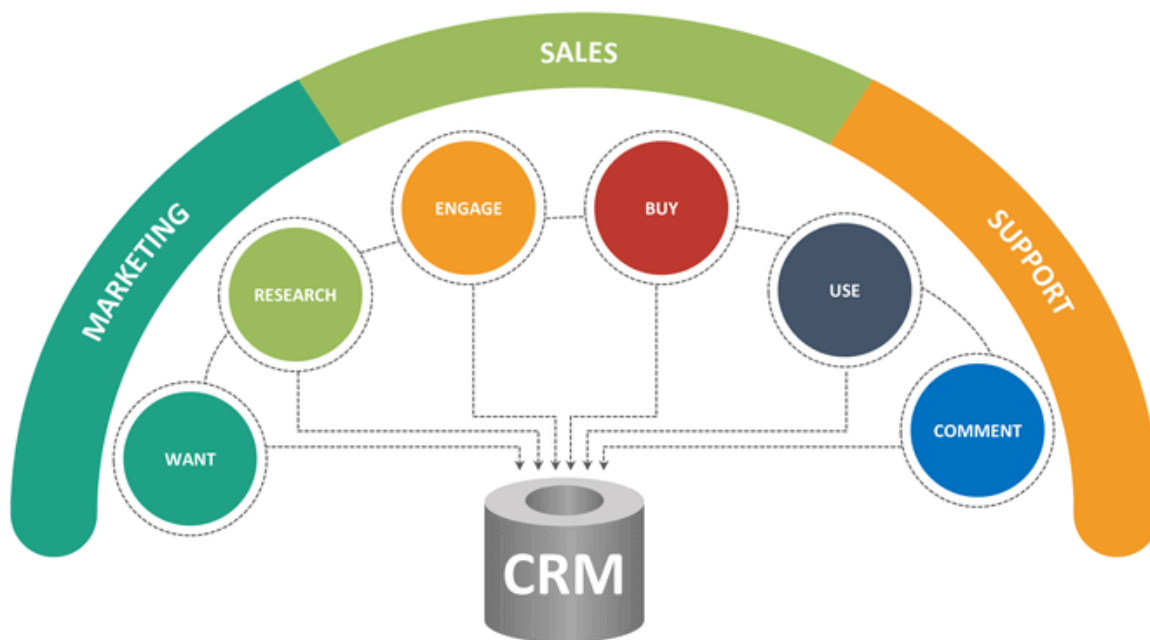


Рис.6 Принцип работы CRM-системы

По первым данным предполагалось, что причиной может быть перегрузка самого приложения, однако дальнейшее исследование показало — **задержки возникают на уровне базы данных, особенно при выполнении тяжёлых SELECT-запросов.**

Я подключался к базе через предоставленный интерфейс мониторинга и проверял список "медленных" запросов, обращая внимание на их частоту, объём выборки и индексацию. Также анализировал отдельные службы CRM на наличие внутренней нагрузки и коллизий с кэшем.

Временное решение включало перезапуск службы, чтобы восстановить отклик.

После этого я продолжал наблюдение и собирал статистику по повторяющимся случаям. Этот опыт научил меня важности оптимизации работы с базой данных и того, как даже одна плохо составленная выборка может сказаться на скорости всей системы

3.7 Работа с API и тестированием эндпоинтов

Одна из наиболее интересных и приближённых к программированию задач во время практики касалась тестирования **API** — внутреннего интерфейса, через который внешние и внутренние компоненты компании обмениваются данными. Это **API** используется, например, в CRM, админ-панели и мобильных решениях.

Я подключился к тестовой среде **Swagger**, которая предоставляла удобный web-интерфейс для взаимодействия с доступными эндпоинтами. Передо мной стояла задача протестировать часть новых методов, добавленных в рамках расширения функционала платформы. Это были, в частности:

- 1) Эндпоинты для получения обновленной статистики пользователя.
- 2) Обновление настроек профиля.
- 3) Вспомогательные методы для работы с уведомлениями.
- 4) Выгрузка последней статистики в **.xml** формате

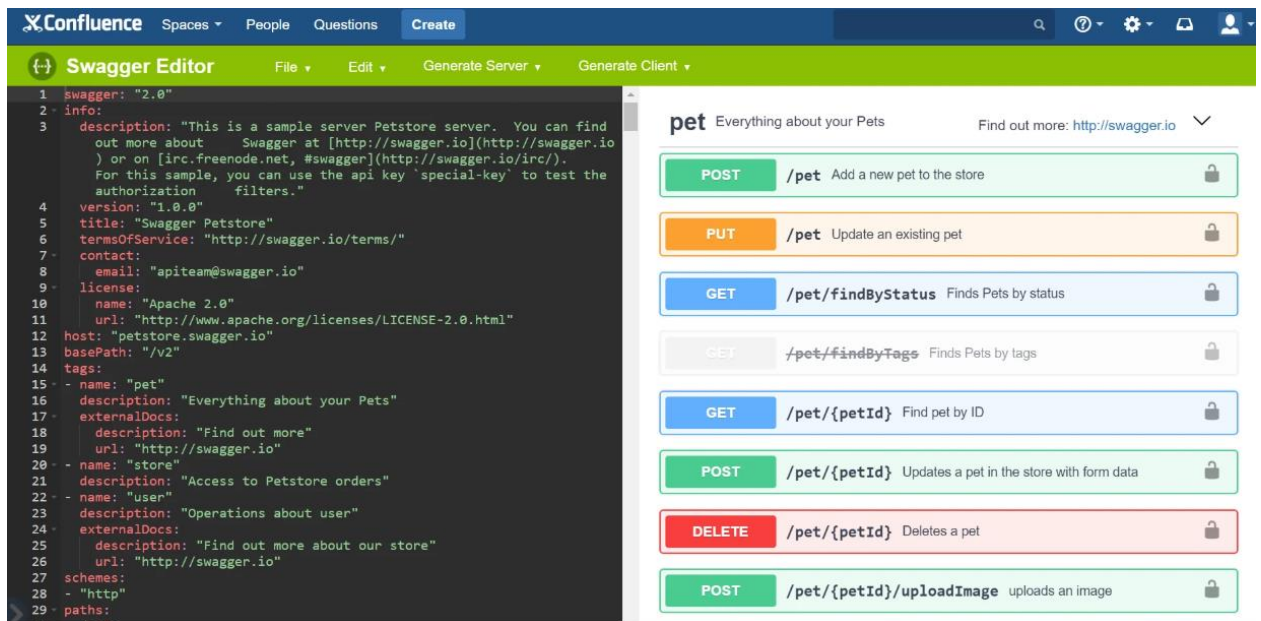


Рис.6 Графический интерфейс Swagger

Что именно было сделано:

- 1) Изучал документацию каждого эндпоинта: какие поля требуются, сначала внимательно какие методы (GET, POST, PUT), какие возможны ошибки.
- 2) Затем выполнял реальные запросы через **Swagger**, проверяя, корректно ли обрабатываются входные параметры и как система реагирует на граничные случаи.
- 3) При обнаружении неточностей (например, неочевидные ошибки валидации) — я фиксировал это в виде отчета с указанием точного запроса и полученного ответа.
- 4) В нескольких случаях, когда эндпоинт возвращал непредсказуемые значения (например, пустые поля без ошибок), я документировал поведение и информировал руководителя.

Отдельно разобрал как устроены **Swagger**-файлы: узнал о формате **OpenAPI**, который описывает структуру **API** в виде **JSON/YAML**. Это дало мне представление о том, как документация генерируется автоматически из кода, и какие требования предъявляются к описанию полей и типов данных.

Под конец практики я повторно тестировал все обновлённые эндпоинты в стабильной тестовой среде. Это включало проверку уже исправленных проблем, а также общее тестирование, чтобы убедиться, что обновления не вызвали побочных эффектов. Вся работа по тестированию была задокументирована, и финальный отчёт был передан руководителю практики.

4. Полученные знания и навыки

В течение практики мне удалось поучаствовать в широком спектре задач, связанных с сопровождением ИТ-инфраструктуры. Несмотря на вспомогательный характер многих из них, они позволили мне получить практическое представление о реальной работе ИТ-отдела в компании.

В результате я освоил или углубил знания в следующих областях:

- 1) Принципы работы корпоративной сети и взаимодействия сервисов.
- 2) Работа с тикет-системой и обработка пользовательских заявок.
- 3) Настройка и подготовка рабочих станций, установка ОС и прикладного ПО.
- 4) Основы обслуживания серверов и работы с RAID-массивами (утилита mdadm).
- 5) Мониторинг состояния системы через **Grafana**, чтение и интерпретация дашбордов.
- 6) Диагностика проблем производительности, в том числе работа с логами и анализ задержек в БД.
- 7) Написание небольших вспомогательных скриптов на Python для автоматизации анализа логов.
- 8) Работа с API и инструментом **Swagger**: тестирование, валидация, оформление отчётов.
- 9) Понимание структуры и логики **OpenAPI** (swagger-файлов) и их роли в документации.

Практика позволила мне научиться быстро разбираться в новых инструментах, работать с документацией, формировать технические отчёты и быть частью командной работы даже при частичном участии в задачах.

5. Заключение

Производственная практика стала для меня важным этапом в профессиональном развитии. Я получил возможность применить теоретические знания на практике, познакомиться с инструментами и технологиями, которые действительно используются в ИТ-инфраструктуре современной компании.

Особенно ценным оказался опыт участия в диагностике и устранении реальных инцидентов — таких, как сбои в работе сайта и снижение производительности CRM. Также полезной была работа с API, которая позволила глубже понять механику клиент-серверного взаимодействия и требования к качественной документации.

Благодаря практике я стал увереннее в вопросах базового администрирования, мониторинга и тестирования. Полученные навыки, безусловно, пригодятся мне как в будущей учебной деятельности, так и в профессиональной работе.