|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА**  **Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Обоснование и разработка требований к программным системам»

Практическое занятие № 7

(вариант №13)

|  |  |
| --- | --- |
| Студент группы | *ИКБО-11-22, Гришин Андрей Валерьевич* |
|  | (подпись) |
| Преподаватель | *Ахмедова Хамида Гаджиалиевна* |
|  | (подпись) |
| Отчет представлен | « » 2024\_г. |

Москва 2024 г.

**Цель работы:** научиться разрабатывать нефункциональные требования к системе.

# Задачи:

* описание требований к качеству продукта;
* описание ограничений системы;
* описание характеристик внешнего интерфейса.

**Используемое ПО:** Draw.io

# Постановка задачи

1. Опишите нефункциональные требования к программной системе, а именно требования к качеству в виде текста в зависимости от типа проекта.
2. Опишите в виде требований накладываемые внешними заинтересованными лицами, существующими соглашениями, решениями руководства и техническими решениями или системами, взаимодействующими с создаваемой системой ограничения программной системы.
3. Опишите требования к внешним интерфейсам:
   * опишите пользовательский интерфейс в виде карты диалоговых окон или создайте общую схему общего взаимодействия прототипов экранных форм;
   * разработайте две диаграммы: диаграмму компонентов и диаграмму размещения. Дайте характеристики каждого интерфейса между компонентами ПО и оборудования системы. В описание могут входить типы поддерживаемых устройств, взаимодействия данных и элементов управлений между ПО и оборудованием, а также протоколы взаимодействия, которые будут использоваться. По возможности перечислите входные и выходные данные, их формат, разрешенные значения или их

диапазоны, а также все временные характеристики, о которых должны знать разработчики.

* + опишите коммуникационные интерфейсы в виде текста. Укажите требования для любых функций взаимодействия, которые будут использоваться продуктом, включая электронную почту, веб-браузер, сетевые протоколы и электронные формы. Определите соответствующие форматы сообщений. Опишите особенности безопасности взаимодействия или шифрования, скорости передачи данных и механизмов согласования и синхронизации. Укажите все ограничения этих интерфейсов, например, допустимость тех или иных типов вложений в сообщениях электронной почты.

1. Требования к качеству программной системы:

**Внешние атрибуты качества:**

1. Доступность
   * AVL-1: Система должна быть доступна не менее 99,8% времени в течение месяца.
   * AVL-2: Плановые работы допускаются только в ночное время (с 02:00 до 05:00) и не чаще двух раз в месяц.
   * AVL-3: Среднее время восстановления после сбоя не должно превышать 15 минут.
2. Производительность
   * PER-1: Время отклика интерфейса на запрос пользователя не более 1,5 секунд.
   * PER-2: Поддержка одновременной работы до 200 пользователей без снижения производительности.
   * PER-3: Обработка до 10 000 объектов данных в день.
3. Безопасность
   * SEC-1: Двухфакторная аутентификация для всех пользователей.
   * SEC-2: Шифрование данных при передаче (протокол TLS 1.3).
   * SEC-3: Логирование всех действий, связанных с изменением критически важных данных.
   * SEC-4: Разграничение прав доступа в зависимости от ролей пользователей.
4. Удобство использования
   * USE-1: Интуитивно понятный интерфейс, доступный для понимания пользователем без предварительного обучения.
   * USE-2: Наличие интерактивных подсказок и обучающих материалов прямо в системе.
   * USE-3: Поддержка корректной работы на современных версиях популярных браузеров (Google Chrome, Mozilla Firefox, Safari).
5. Надёжность
   * REL-1: Среднее время между отказами (MTBF) должно составлять не менее 180 дней.
   * REL-2: Система должна автоматически сохранять критически важные данные каждые 5 минут.
6. Совместимость
   * IOP-1: Программа должна корректно работать на операционных системах Windows 10, macOS 11.0 и Linux.
   * IOP-2: Поддержка стандартных форматов данных для обмена: JSON, XML.

**Внутренние атрибуты качества:**

1. Эффективность
   * EFF-1: На пике нагрузки приложение не должно потреблять более 75% процессорного времени и 50% оперативной памяти.
2. Масштабируемость
   * SCA-1: Возможность увеличения нагрузки до 300% в течение одного года без изменения архитектуры системы.
3. Модифицируемость
   * MOD-1: Обновление функциональности должно быть возможно без изменения основного кода системы более чем на 15%.
4. Переносимость
   * POR-1: Возможность миграции системы на другой сервер без потери данных или необходимости значительных изменений.
5. Проверяемость
   * VER-1: Каждый модуль системы должен иметь отдельные тесты, которые можно выполнить независимо от других модулей.
6. Повторное использование
   * REU-1: Функции, обрабатывающие данные, должны быть реализованы в виде библиотек, которые можно повторно использовать в других проектах.
7. Ограничения программной системы: Технические ограничения:
   * CON-1: Поддержка только современных веб-браузеров с актуальными версиями (не старше двух лет).
   * CON-2: Использование реляционной СУБД PostgreSQL версии не ниже 13.
   * CON-3: Необходима поддержка HTTPS протокола
   * CON-4: Необходимость развертывания системы на серверах с минимальными характеристиками: 4 ядра CPU, 16 ГБ RAM, SSD-хранилище.

Организационные ограничения:

* + CON-5: Соответствие требованиям GDPR по защите данных
  + CON-6: Соблюдение корпоративных стандартов кодирования
  + CON-7: Все сторонние библиотеки и фреймворки должны иметь лицензии, совместимые с корпоративными стандартами.
  + CON-8: Регулярный аудит безопасности не реже одного раза в год.

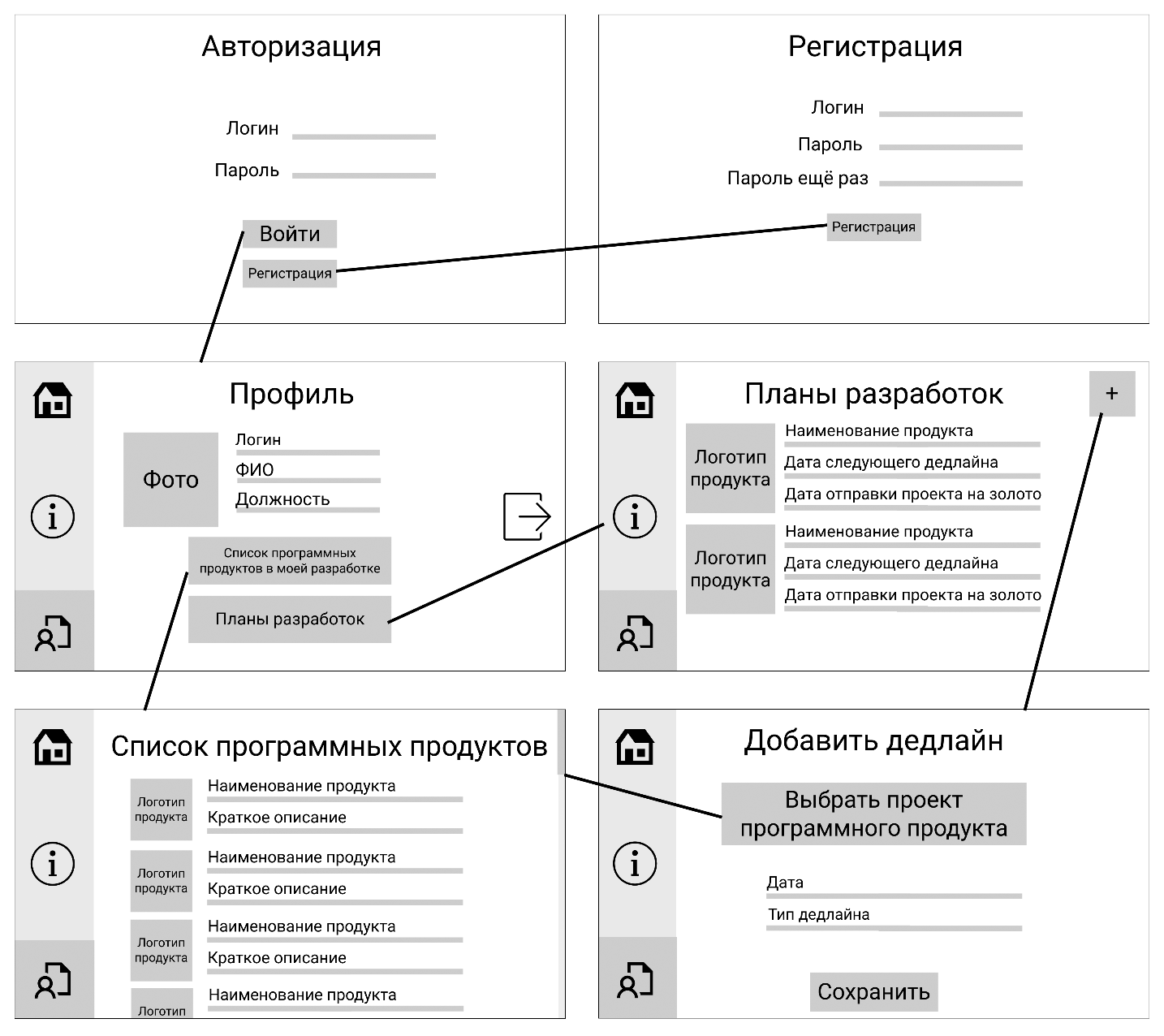


Рисунок 1 – Взаимодействие прототипов форм

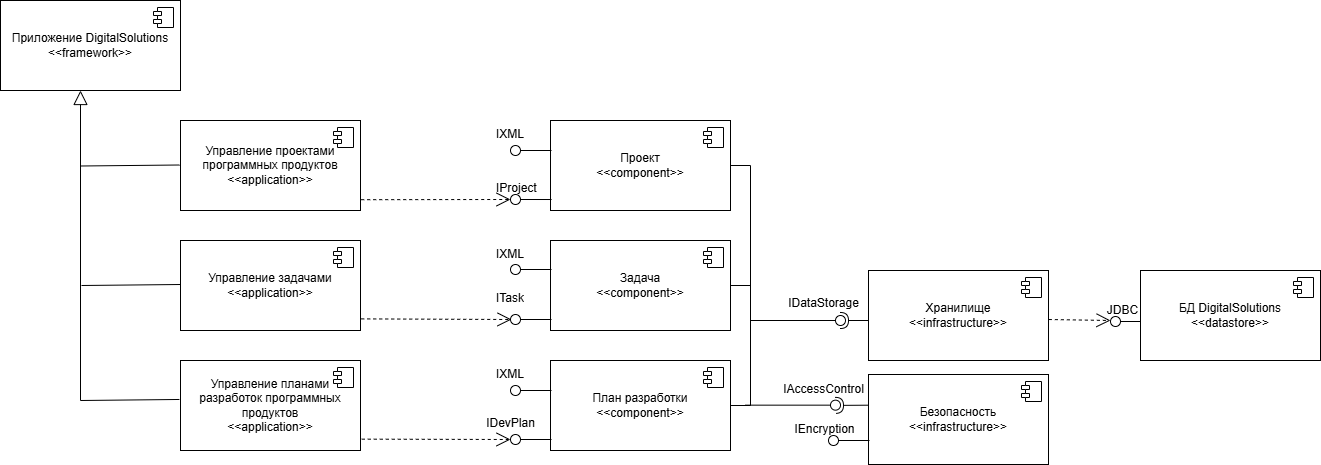


Рисунок 6 – Диаграмма компонентов

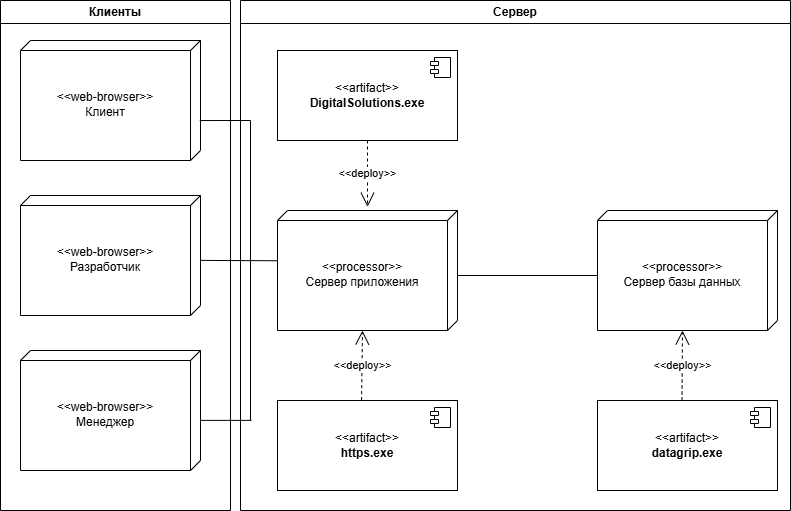


Рисунок 7 – Диаграмма размещения

Коммуникационные интерфейсы для системы управления разработками цифровой компании:

1. Сетевые протоколы:
   * **HTTPS (TLS 1.3):** Защищенная передача данных между клиентом и сервером.
   * **WebSocket:** Для двусторонней связи в режиме реального времени.
   * **HTTP/3:** Оптимизация скорости взаимодействия между клиентом и сервером.
   * **SMTP/SMTPS:** Для отправки email-уведомлений о статусах задач и проектах.
   * **OAuth 2.0:** Для обеспечения безопасной аутентификации пользователей.
2. Форматы обмена данными:
   * **XML:** Основной формат для передачи данных между внутренними и внешними системами.
   * **JSON:** Для легковесных API-запросов, используемых клиентами.
   * **CSV:** Для экспорта данных.
   * **PDF:** Для генерации документов
3. Безопасность коммуникаций:
   * **SSL/TLS шифрование:** Защита всех передаваемых данных (длина ключа 256 бит)
   * **JWT (JSON Web Token):** Для авторизации и передачи пользовательских прав.
   * **Двухфакторная аутентификация:** Для повышения уровня безопасности.
   * **Механизмы защиты от DoS-атак:** Ограничение количества запросов от одного пользователя.
   * **Подпись и проверка целостности XML-документов:** Для защиты от несанкционированных изменений данных.
4. Интеграционные интерфейсы:
   * **REST API:** Для интеграции с внешними системами управления проектами.
   * **SOAP API:** Для взаимодействия с корпоративными системами, где требуется повышенная надежность.
   * **Webhook:** Для отправки событийных уведомлений
   * **OpenAPI спецификация:** Для документирования всех доступных REST API.
   * **GraphQL:** Для выборочного получения данных по сложным запросам.
5. Характеристики передачи:
   * **Максимальный размер payload:** 15 МБ.
   * **Время отклика сервера:** Не более 300 мс.
   * **Пропускная способность:** До 150 Мбит/с.
   * **Сжатие данных:** Поддержка gzip для уменьшения объема передаваемых данных.
6. Ограничения интерфейсов:
   * **Поддерживаемые форматы вложений:** XML, JSON, CSV, PDF, PNG, JPG, DOC, DOCX.
   * **Максимальный размер файлов для загрузки:** 10 МБ.
   * **Максимальное время установки соединения:** 4 секунды.
   * **Автоматический разрыв соединений:** Через 20 минут неактивности.
7. Механизмы синхронизации:
   * **NTP:** Для синхронизации времени между сервером и клиентами.
   * **Очереди задач:** Использование брокеров сообщений (например, RabbitMQ или Kafka) для управления обработкой данных.
   * **Кэширование:** Снижение нагрузки на сервер через использование Redis.
   * **Репликация базы данных:** Для обеспечения высокой доступности и отказоустойчивости.
   * **Синхронизация XML-документов:** Механизм контроля версий и слияния изменений.

# Список использованных источников и литературы:

1. Вигерс Карл, Битти Джой. Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М. : Издательство

«Русская редакция» ; СПб. : БХВ-Петербург, 2014 — 736 стр. : ил.

1. Остроух, А.В. Интеллектуальные информационные системы и технологии: монография / А.В. Остроух, А.Б. Николаев. — Санкт-Петербург : Лань, 2019.

— 308 с. — ISBN 978-5-8114-3409-1. —URL: https://e.lanbook.com/book/115518

1. Зубкова, Т.М. Технология разработки программного обеспечения : учебное пособие / Т.М. Зубкова. — Санкт-Петербург : Лань, 2019. — 324 с. — ISBN 978-5-8114-3842-6. — URL: https://e.lanbook.com/book/122176
2. Иванова Г. С. Технология программирования: Доп. УМО в кач. учебника для вузов – М.:Кнорус, 2013. — 333 с.: ил.
3. Методология и технология проектирования информационных систем : учебное пособие / Ю.М. Казаков, А.А. Тищенко, А.А. Кузьменко [и др.]. — Москва : ФЛИНТА, 2018. — 136 с. — ISBN 978-5-9765-4013-2. — URL:

https://e.lanbook.com/book/113460