**ПРАКТИЧЕСКАЯ РАБОТА №2**

Тема: Разработка требований к программному продукту.

Цель: изучить процесс разработки требований; выполнить анализ требований к разрабатываемому программному продукту; оформить техническое задание на разработку программного обеспечения

**Задание:**

1. Информационная модель будущего продукта, включающую в себя описание основных объектов системы и взаимодействия между ними.

Общее описание системы.

Предполагается разработка программного средства, которое обеспечивает автоматизированный сбор требований, проектирование, разработку модулей, их тестирование и внедрение. Система должна поддерживать жизненный цикл разработки программного обеспечения и обеспечивать прозрачность, управление задачами и отчетность.

Основные объекты системы и их описание.

Основные объекты системы и их описание представлены в таблице 1.

Таблица 1 – Основные объекты системы и их описание

|  |  |
| --- | --- |
| Объект | Описание |
| Требование | Представляет собой функциональные и нефункциональные требования к продукту. |
| Проект | Единая сущность, объединяющая задачи, модули и участников разработки. |
| Задача | Декомпозированная единица работы по проектированию, разработке, тестированию. |
| Модуль | Независимая часть продукта, разрабатываемая отдельно и интегрируемая в проект. |
| Тестовый сценарий | Набор условий для проверки корректной работы модуля или системы. |
| Отчет | Документ с результатами выполнения этапов: тестирования, разработки, сборки. |
| Пользователь | Участник системы (роль: Аналитик, Разработчик, Тестировщик, Менеджер). |
| Продолжение таблицы 1 | |  |
| Объект | Описание |
| Взаимодействие | Процесс коммуникации между пользователями и объектами системы. |

Взаимодействие объектов.

Пользователь взаимодействует с системой через интерфейс:

* Аналитик добавляет требования к проекту.
* Разработчик разбивает требования на задачи и разрабатывает модули.
* Тестировщик создает тестовые сценарии, проверяет модули и создает отчеты.
* Менеджер управляет статусами проектов и задач, следит за прогрессом.

Проект объединяет:

* Требования, которые детализируются в задачи.
* Модули, созданные в процессе разработки.
* Тестовые сценарии и отчеты, которые проверяют качество продукта.

Задача связана:

* С конкретным требованием и модулем.
* Обновляет статусы в системе: "Назначена", "Выполняется", "Готова к тестированию".

Тестовый сценарий запускается на модуле.

* Результат тестирования фиксируется в отчете.

Диаграмма взаимодействия.

UML-диаграмма взаимодействия объектов системы представлена на рисунке 1.



Рисунок 1 – UML-диаграмма взаимодействия объектов системы

1. Системные требования для разработки и эксплуатации продукта.

Функциональные требования.

Сбор требований

* Система должна позволять добавлять, редактировать и удалять требования к проекту.
* Поддержка версионности требований.

Управление проектами и задачами

* Возможность создания, управления и отслеживания статуса проектов.
* Декомпозиция требований на задачи с назначением ответственных.
* Возможность установки приоритетов задач и сроков исполнения.

Разработка модулей

* Возможность загрузки и хранения кода модулей.
* Связь модулей с задачами и требованиями.

Тестирование

* Автоматизированный запуск тестов с отчетами о прохождении.
* Создание и хранение тестовых сценариев.

Отчеты и аналитика

* Генерация отчетов о статусе задач, прогрессе проекта и результатах тестирования.
* Визуализация данных (графики, таблицы).

Роли и доступы

* Поддержка ролей: Аналитик, Разработчик, Тестировщик, Менеджер.
* Ограничение доступа к функциям на основе ролей.

Нефункциональные требования.

Производительность

* Система должна обрабатывать до 1000 задач одновременно без снижения скорости работы.

Масштабируемость

* Возможность масштабирования для поддержки большого количества проектов и пользователей.

Безопасность

* Аутентификация и авторизация пользователей.
* Шифрование данных.

Доступность

* Система должна быть доступна 24/7 с временем простоя не более 2 часов в месяц.

Интеграция

* Возможность интеграции с системами контроля версий (Git) и CI/CD (Jenkins).

Требования к эксплуатации.

* Система должна иметь веб-интерфейс с поддержкой всех современных браузеров.
* Обновления должны быть возможны без остановки работы системы.
* Поддержка логирования и мониторинга ошибок.

1. Разработка технического задания.

Вариант 5.

Тема: Приложение для просмотра и изменения расписания.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программного продукта

«\_\_\_\_\_\_\_\_»

РАЗРАБОТЧИК

Зеневич Александр Олегович\_\_

ФИО

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

СОГЛАСОВАНО

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Дата изменения |
|  |  |  |  |
|  |  |  |
|  |  |  |

Минск – 2024

Содержание

[1 Общие сведения 7](#_Toc185382575)

[1.1 Наименование программного продукта 7](#_Toc185382576)

[1.2 Назначение и область применения 7](#_Toc185382577)

[1.3 Целевая аудитория продукта – пользователи 7](#_Toc185382578)

[2 Общие требования к программному продукту 8](#_Toc185382579)

[2.1 Требования к функциональным характеристикам 8](#_Toc185382580)

[2.2 Взаимосвязь программы с другими информационными системами 8](#_Toc185382581)

[2.3 Требования к лингвистическому обеспечению 8](#_Toc185382582)

[3 Системные требования (условия эксплуатации) 9](#_Toc185382583)

[3.1 Требования к составу и параметрам технических средств 9](#_Toc185382584)

[3.2 Требования к информационной и программной совместимости 9](#_Toc185382585)

[3.2.1 Требования к информационным структурам и методам решения 9](#_Toc185382586)

[3.2.2 Требования к исходным кодам и языкам программирования 9](#_Toc185382587)

[3.2.3 Требования к программным средствам, используемым программой 9](#_Toc185382588)

[3.2.4 Требования  к защите информации и программ 9](#_Toc185382589)

[4 Требования к интерфейсу 11](#_Toc185382590)

[4.1 Требования к внешнему интерфейсу 11](#_Toc185382591)

[4.2 Графический интерфейс пользователя 11](#_Toc185382592)

# Общие сведения

## Наименование программного продукта

Программный продукт: «Приложение для просмотра и изменения расписания»

## Назначение и область применения

Программа предназначена для просмотра и редактирования расписаний занятий, встреч или других мероприятий. Может использоваться в образовательных учреждениях, офисах, а также индивидуальными пользователями для управления своим временем.

## Целевая аудитория продукта – пользователи

Приложение предназначено для использования:

* студентами и преподавателями;
* офисными сотрудниками и менеджерами;
* индивидуальными пользователями для личного планирования.

# Общие требования к программному продукту

## Требования к функциональным характеристикам

Требования к функциональным характеристикам:

* Просмотр текущего расписания в удобном формате (таблица, календарь);
* Редактирование расписания с возможностью добавления, удаления и изменения записей;
* Уведомления о предстоящих событиях;
* Импорт и экспорт расписания в популярных форматах (CSV, iCal);
* Поиск и фильтрация записей по различным параметрам.

## Взаимосвязь программы с другими информационными системами

Программа должна поддерживать интеграцию с Google Calendar, Microsoft Outlook и другими популярными сервисами.

## Требования к лингвистическому обеспечению

Программа должна поддерживать работу на русском и английском языках. Язык интерфейса выбирается пользователем при первом запуске или в настройках.

# Системные требования (условия эксплуатации)

## Требования к составу и параметрам технических средств

Требования к составу и параметрам технических средств:

* Операционная система: Windows 10 и выше, macOS 10.15 и выше, Android 9.0 и выше, iOS 13 и выше;
* Процессор: не менее 2 ГГц;
* Оперативная память: не менее 4 ГБ;
* Дисковое пространство: не менее 500 МБ.

## Требования к информационной и программной совместимости

## Требования к информационным структурам и методам решения

Программа должна использовать стандартные форматы данных (JSON, XML) для взаимодействия с другими системами.

## Требования к исходным кодам и языкам программирования

Требования к исходным кодам и языкам программирования:

* Языки программирования: C# для серверной части, TypeScript (Angular) для клиентской части;
* Исходный код должен быть оформлен в соответствии со всеми актуальными стандартами.

## Требования к программным средствам, используемым программой

Программа должна работать автономно или с использованием серверной части на базе Docker-контейнеров.

## Требования к защите информации и программ

Требования к защите информации и программ:

* Защита данных пользователя с помощью шифрования (TLS);
* Защита от несанкционированного доступа через систему авторизации и аутентификации.

# Требования к интерфейсу

## Требования к внешнему интерфейсу

Программа должна взаимодействовать с:

* локальной файловой системой для экспорта и импорта данных;
* удаленными API для синхронизации расписания.

## Графический интерфейс пользователя

Графический интерфейс пользователя:

* Поддержка адаптивного дизайна для работы на устройствах с различными размерами экранов;
* Интуитивно понятная навигация с использованием стандартных элементов интерфейса (кнопки, вкладки, выпадающие списки);
* Возможность настройки внешнего вида (тема, шрифты).