Содержание

[Введение 2](#_Toc189662038)

[1. Теоретическая часть 4](#_Toc189662039)

# Введение

В современном мире командная работа стала неотъемлемой частью образовательной и профессиональной деятельности. Эффективное взаимодействие между участниками группы требует удобных инструментов для общения, распределения задач и отслеживания. Однако многие существующие решения либо сложны в освоении для обычных пользователей, либо не обладают достаточной функциональностью для полноценного управления групповой работой.

Программное обеспечение для организации работы в групповых проектах должно обеспечивать удобные средства коммуникации, позволяя пользователям вести обсуждения в общем чате и личных сообщениях. Важной частью является система управления задачами, где каждый участник может фиксировать свою работу, следить за прогрессом коллег и устанавливать дедлайны.

Кроме того, важной функцией такого программного обеспечения является обеспечение прозрачности и подотчётности внутри команды. Возможность отслеживать задачи каждого участника и видеть прогресс работы позволяет значительно улучшить координацию, предотвратить недоразумения и повысить общую продуктивность группы. Такие инструменты также стимулируют участие каждого члена команды в процессе, а уведомления о предстоящих сроках в проекте помогают избежать задержек и пропуска важной информации.

Актуальность данной темы обусловлена возрастающей потребностью в удобных и эффективных инструментах для организации командной работы. Разработка программного обеспечения, учитывающего современные требования пользователей, способствует оптимизации рабочих процессов и повышению качества выполнения групповых проектов.

Цель: Разработка программного обеспечения для организации работы в групповых проектах.

Задачи:

1. Составление технического задание.
2. Проанализировать и обосновать выбор инструментария.
3. Провести анализ предметной области.
4. Создать макет будущего программного обеспечения.
5. Разработать программное обеспечение.
6. Провести тестирование.
7. Написать инструкцию по инсталляции и эксплуатации.

1. Теоретическая часть

1.1 Техническое задание

1. Наименование программного продукта или программы:

«GroupTasker».

2. Краткая характеристика области применения: организация и управление работы в групповых проектах.

3. Основание для разработки

3.1 Основания для проведения разработки: Задание курсовой работы от преподавателя ГБПОУ НСО «Новосибирский химико-технологический колледж им. д. и. Менделеева».

3.2 Наименование и условное обозначение разработки: Программное обеспечение для работы в группах «GroupTasker».

4. Назначение разработки

4.1 Функциональное назначение:

Обеспечение доступа к организации и управлении работы в групповых проектах, создавать группы, писать на доске заметок, переписывать в чатах и расписывать свои задачи.

4.2 Эксплуатационное назначение: для использования приложения пользователями.

5. Требования к функциональным характеристикам.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

6. Состав выполняемых функций:

* Функции главного меню:
* Функция кнопки "Создать группу" позволяет перейти на страницу создания новой группы.
* Функция кнопки "Войти в группу" позволяет перейти на страницу входа в существующую группу.
* Функции окна создания группы:
* Функция кнопки "Back" позволяет вернуться на главную страницу приложения.
* Функция полей ввода позволяет пользователю ввести название группы, свое имя и пароль.
* Функция кнопки "Создать группу" создает новую группу с указанными параметрами и генерирует уникальный код группы.
* Функция автоматического копирования кода группы после создания упрощает обмен кодом между участниками.
* Функции окна входа в группу:
* Функция кнопки "Back" позволяет вернуться на главную страницу приложения.
* Функция полей ввода позволяет пользователю ввести код группы, свое имя и пароль.
* Функция кнопки "Присоединиться к группе" проверяет данные и добавляет пользователя в группу.
* Функции главного окна приложения:
* Функция кнопки "Back" позволяет вернуться на главную страницу приложения.
* Функция бокового меню позволяет переключаться между разделами приложения.
* Функция кнопки "Группа" открывает панель управления группой.
* Функция кнопки "Мои задачи" открывает доску заметок группы.
* Функция отображения имени пользователя в верхней панели показывает текущего пользователя.
* Функции панели управления группой:
* Функция отображения названия группы показывает информацию о текущей группе.
* Функция списка участников отображает всех членов группы.
* Функция кнопки "Home" позволяет вернуться к доске заметок.
* Функция кнопки "Исключить участника" (для создателя группы) позволяет удалить участника из группы.
* Функция кнопки "Удалить группу" (для создателя группы) позволяет полностью удалить группу.
* Функции доски заметок:
* Функция кнопки "Добавить заметку" открывает диалоговое окно для создания новой заметки.
* Функция отображения заметок показывает все заметки группы в хронологическом порядке.
* Функция подписи автора у каждой заметки указывает, кто создал заметку.
* Функция кнопки удаления позволяет удалить заметку.
* Функция автоматического обновления списка заметок после изменений.
* Дополнительные функции:
* Функция валидации ввода проверяет заполнение обязательных полей.
* Функция отображения сообщений об ошибках информирует пользователя о проблемах.
* Функция стилизации интерфейса обеспечивает единый визуальный стиль приложения.

7. Требования к видам обслуживания:

Для поддержки работоспособности приложения GroupTasker требуется системный администратор.

8. Требования к квалификации пользователя:

Для работы с приложением пользователю необходимо базовое знание ПК на уровне пользователя.

9. Привлечение пользователей:

* Простота интерфейса способствует быстрому освоению приложения.
* Возможность совместной работы в группе повышает полезность продукта.
* Минимальные требования к оборудованию делают приложение доступным.
* Система копирования кода группы упрощает приглашение новых участников.

10. Типы поддерживаемых платформ:

Windows 10/11.

11. Минимальные системные требования:

* Процессор: 1 ГГц или выше
* Оперативная память: 2 ГБ
* Дисковое пространство: 1 ГБ
* Наличие интернет-соединения (мин. 5 Мбит/с)

12. Требования к исходным кодам и языкам программирования:

* Python 3.11
* PyQt6 (для графического интерфейса)
* PostgreSQL (для работы с базой данных)
* psycopg2 (для подключения к БД)

13. Требования к программным средствам:

* Интерпретатор Python
* Библиотека PyQt6
* Система управления базами данных PostgreSQL
* Менеджер пакетов pip

14. Стадии разработки:

* Планирование
* Проектирование
* Разработка
* Тестирование
* Внедрение

15. Этапы и содержание работы:

16. Сроки разработки:

Общий срок разработки проекта: 10 недель.

17. Календарный план работ:

18. Виды испытаний:

* Проверка работы всех кнопок и элементов управления
* Тестирование создания и удаления заметок
* Проверка работы с группами (создание, вход, удаление)
* Проверка корректности работы базы данных

19. Общие требования к приёму работы:

Программный продукт считается готовым, если:

* Все запланированные функции реализованы
* Интерфейс работает корректно
* Обеспечена базовая безопасность данных
* Написана сопроводительная документация

20. Порядок контроля и приемки:

* Демонстрация работоспособности основных функций
* Проверка соответствия техническому заданию
* Тестирование на различных платформах
* Проверка документации

1.2. Описание инструментария и его обоснования

1.2.1 Выбор языка программирования

Язык программирования является фундаментальным инструментом для создания программного обеспечения. Среди множества существующих языков (более 300) для реализации данного проекта требовалось выбрать наиболее подходящий, способный обеспечить выполнение всех поставленных задач. Был проведен сравнительный анализ современных языков программирования.

Python — интерпретируемый высокоуровневый язык общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Основные особенности:

* Ориентирован на повышение производительности разработчика
* Обеспечивает высокую читаемость кода
* Поддерживает несколько парадигм программирования
* Имеет обширную стандартную библиотеку
* Обеспечивает переносимость программ

Для данного проекта Python был выбран благодаря:

1. Простоте создания графических интерфейсов (библиотека PyQt6)
2. Удобству работы с базами данных (psycopg2 для PostgreSQL)
3. Кроссплатформенности
4. Быстроте разработки
5. Большому количеству специализированных библиотек

Среди альтернативных языков рассматривались:

C++ — компилируемый статически типизированный язык общего назначения. Основные характеристики:

* Поддерживает процедурное, ООП и обобщенное программирование
* Имеет богатую стандартную библиотеку
* Позволяет низкоуровневый доступ к памяти
* Требует ручного управления ресурсами
* Сложный синтаксис по сравнению с Python

Области применения C++:

* Системное программирование
* Разработка игр
* Высоконагруженные серверы
* Встраиваемые системы

Java — компилируемый объектно-ориентированный язык со строгой типизацией. Особенности:

* Исполняется на виртуальной машине JVM
* Автоматическое управление памятью
* Богатая стандартная библиотека
* Менее удобен для создания GUI чем Python+Qt

JavaScript/TypeScript — основные языки веб-разработки. Характеристики:

* Динамическая типизация (статическая в TypeScript)
* Асинхронная модель выполнения
* Огромная экосистема фреймворков
* Оптимальны для веб-приложений
* Менее подходят для десктопных программ

Для данного проекта Python показал себя как оптимальный выбор, сочетающий простоту разработки, богатые возможности для создания GUI и работы с БД, а также кроссплатформенность.

1.2.2 Выбор среды разработки

Интегрированная среда разработки (IDE) — важный инструмент, значительно повышающий продуктивность программиста. Для проекта рассматривались несколько популярных IDE.

PyCharm — специализированная среда для Python-разработки от JetBrains. Основные возможности:

* Полноценная поддержка Python и его экосистемы
* Интеллектуальное автодополнение кода
* Визуальный отладчик с точками останова
* Интеграция с системами контроля версий
* Поддержка научных инструментов
* Инструменты для работы с базами данных
* Возможности рефакторинга

Visual Studio Code — легковесный кроссплатформенный редактор кода. Особенности:

* Поддержка множества языков через расширения
* Встроенный терминал
* Интеграция с Git
* Широкая кастомизация
* Меньшая "из коробки" поддержка Python чем у PyCharm

IDLE — стандартная среда разработки, поставляемая с Python. Характеристики:

* Минималистичный интерфейс
* Базовые возможности редактирования
* Отсутствие продвинутых инструментов
* Нет интеграции с современными системами разработки

Для данного проекта была выбрана среда PyCharm, так как она:

1. Предоставляет наиболее полный набор инструментов именно для Python
2. Имеет отличную поддержку PyQt6
3. Облегчает работу с базами данных
4. Включает мощный отладчик
5. Поддерживает все современные практики разработки

PyCharm значительно ускоряет процесс создания приложений за счет интеллектуальных функций автодополнения, навигации по коду и встроенных инструментов тестирования, что делает его оптимальным выбором для данного проекта.

1.2.3 Выбор библиотек и модулей

Для реализации всех функций проекта были использованы различные библиотеки и модули, которые значительно расширили возможности программы и позволили реализовать задуманный функционал. К таким модулям и библиотекам относятся:

* PyQt6 - современная библиотека для создания графических интерфейсов. Предоставляет богатый набор виджетов и инструментов для разработки кроссплатформенных приложений с профессиональным интерфейсом. Включает систему сигналов и слотов для обработки событий, поддерживает стилизацию элементов и позволяет создавать сложные многооконные приложения.
* psycopg2 - адаптер PostgreSQL для Python, обеспечивающий надежное взаимодействие с базой данных. Поддерживает все основные функции СУБД, включая транзакции, подготовленные запросы и работу с различными типами данных. Позволяет эффективно выполнять SQL-запросы и обрабатывать результаты.
* random - стандартный модуль Python для работы со случайными числами. Использовался для генерации уникальных кодов групп. Модуль предоставляет криптографически безопасные методы генерации и различные распределения случайных величин.
* pyperclip - простая в использовании библиотека для работы с буфером обмена. Обеспечивает кроссплатформенный доступ к системному буферу, что позволяет удобно копировать и вставлять данные между приложением и другими программами.
* sys - базовый модуль Python, предоставляющий доступ к системным функциям интерпретатора. Использовался для обработки аргументов командной строки и управления завершением работы программы.

1.3 Анализ предметной области

Перед разработкой программного обеспечения для группового управления задачами был проведен детальный анализ существующих решений в данной предметной области. Рассмотрим наиболее популярные аналоги:

Trello - облачный сервис для организации задач по методу Kanban. Основные особенности:

* Визуальное представление задач в виде карточек
* Гибкая система досок и колонок
* Возможность совместной работы
* Интеграция с другими сервисами
* Ограниченные возможности для хранения заметок
* Отсутствие встроенного мессенджера

Asana - профессиональный инструмент управления проектами. Характеристики:

* Иерархическая структура задач
* Гибкая система назначений
* Календарь и временные линии
* Отчеты и аналитика
* Сложный интерфейс для новичков
* Ограниченные бесплатные возможности

Notion - универсальный инструмент для организации работы. Особенности:

* Комбинирование заметок и задач
* Гибкие шаблоны
* Возможность совместного редактирования
* Встроенная база данных
* Высокий порог входа
* Требует времени на настройку

Microsoft Teams - корпоративный коммуникационный хаб. Основные функции:

* Чат и видеоконференции
* Интеграция с Office 365
* Совместная работа над документами
* Сложная система управления задачами
* Ориентирован на крупные организации

На основе проведенного анализа можно выделить ключевые требования к разрабатываемому решению:

1. Простота и интуитивность интерфейса
2. Возможность групповой работы
3. Система заметок и задач в одном месте
4. Гибкие права доступа
5. Кроссплатформенность
6. Минимальные системные требования
7. Локальное хранение данных при необходимости
8. Простота создания и подключения к группам
9. Возможность быстрого обмена информацией
10. Отсутствие сложных настроек