



"Разработка перечня артефактов и протоколов проекта"

Изучение создания и управления артефактами проекта - ключевой навык в разработке программного обеспечения. Артефакты и протоколы обеспечивают структурированный подход к документированию всех этапов жизненного цикла проекта.

Артефакт проекта

Материализованный результат деятельности, документ или продукт, создаваемый в процессе разработки ПО

Протокол проекта

Документ, фиксирующий ход события в проекте: совещания, решения, итоги спринта



Классификация артефактов



Документы управления проектом

- Устав проекта (Project Charter)
- План управления проектом
- Бэклог продукта / спринта
- Протоколы совещаний



Артефакты требований

- Видение продукта (Vision Statement)
- Техническое задание / Пользовательские истории
- Модели процессов (BPMN, Use Case)



Артефакты проектирования

- Диаграммы архитектуры (UML)
- Прототипы интерфейсов (Wireframes)
- Модель данных (ER-диаграмма)

Классификация артефактов



Артефакты реализации

- Исходный код (Source Code)
- Репозиторий кода (Git)
- Инструкция по сборке (Build Scripts)



Артефакты тестирования

- Тест-план (Test Plan)
- Тест-кейсы (Test Cases)
- Отчёты о дефектах (Bug Scripts)



Эксплуатационные артефакты:

- Руководство пользователя (User Manual)
- Руководство администратора (Admin Guide)
- Релизные записи (Release Notes)



Практическое задание: Smart To-Do List

Разработка нового сервиса "Умный список задач" с инновационными функциями для повышения продуктивности пользователей.

Создание и управление задачами

Базовый функционал для добавления, редактирования и удаления задач

Автоматическая категоризация

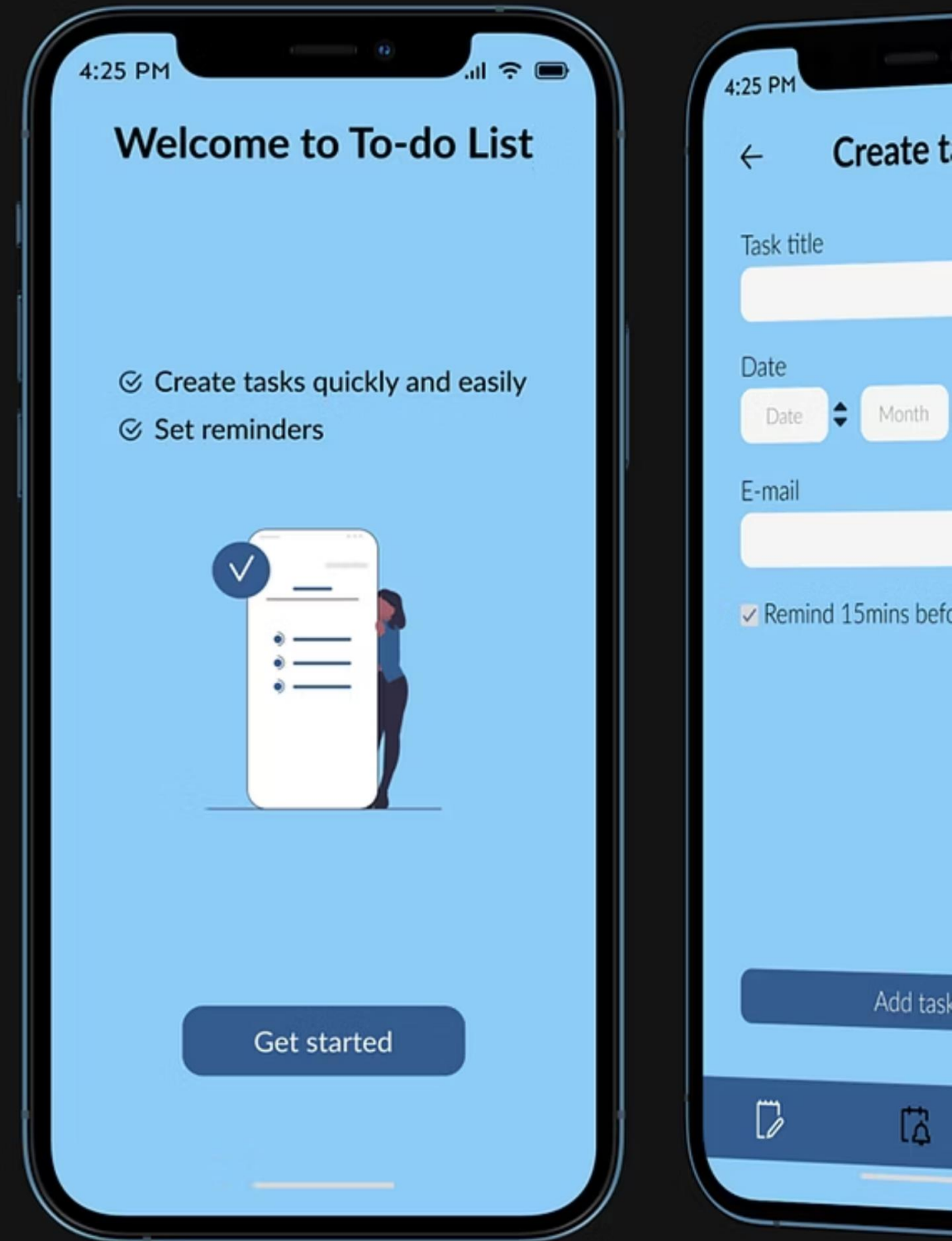
Система анализирует ключевые слова и автоматически назначает категории

Геолокационные напоминания

Уведомления о задачах при приближении к определенным местам

Статистика продуктивности

Визуализация и анализ выполненных задач для повышения эффективности



Sprint Backlog



Задание 1: Документы управления и требований

- 01

Видение продукта

Создание краткого документа с описанием целевой аудитории, решаемых проблем и ключевых преимуществ перед обычными списками задач
- 02

Бэклог продукта

Составление таблицы пользовательских историй в формате: "Как <Роль>, я хочу <Функцию>, чтобы <Ценность>" с приоритизацией
- 03

Протокол совещания

Документирование мозгового штурма по приоритизации бэклога с участниками, решениями и вопросами для проработки

Минимум 8-10 пользовательских историй должны покрывать основные сценарии использования приложения.

Протокол совещания по планированию

Тема: Планирование бэклога продукта для Smart To-Do List

Повестка дня

1. Обсуждение и уточнение пользовательских историй
2. Приоритизация бэклога для первого спринта

Ключевые решения

1. Определение историй для первого спринта: создание, удаление и автоматическая категоризация задач

Вопросы для проработки

Уточнение алгоритма автоматической категоризации и интеграции с геолокационными сервисами





Задание 2: Артефакты проектирования

Прототип интерфейса

Создание low-fidelity wireframe главного экрана с отображением:

- Список задач
- Поле ввода новой задачи
- Кнопка "Добавить"
- Цветные теги категорий

Инструменты: Figma, Draw.io или бумажный набросок

Модель данных

ER-диаграмма основных сущностей:

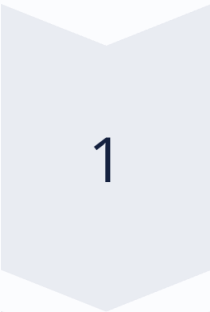
User: id, name, email

Task: id, title, category_id, user_id

Category: id, name, color

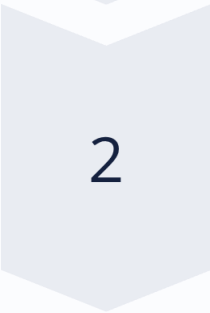
Связи: "один-ко-многим" между сущностями

Задание 3: Реализация и тестирование



Репозиторий кода

Создание публичного репозитория smart-todo-list на GitHub/GitLab с README.md, .gitignore, LICENSE и структурой папок /docs, /src, /tests



Тест-кейсы

Разработка 3-5 тест-кейсов для функции "Создание задачи" в табличном формате с ID, названием, условиями, шагами и ожидаемыми результатами

ID	Название кейса	Шаги	Результат
TC1	Создание задачи с валидным названием	Ввод текста, нажатие "Добавить"	Задача появляется в списке
TC2	Создание задачи с пустым полем	Пустое поле, нажатие "Добавить"	Сообщение об ошибке

Project Status Report Template

This document is your guide to creating a regular project status report. Adapt the doc's inventory depending on who you're writing the report for. Try to give as much information as possible, write clearly, and celebrate each milestone you hit. 🎉

Project name: Project awesome
Project manager: John Smith
Project code: If applicable
Report date: MM/DD/YYYY

Overall project status 🕒
On-track / At-risk / Off-track

Top Level Summary 📄
In a few sentences, give an overview of the project status. Adapt this depending on who the report is for.

Project Components Overview	Owners & Team	Comments
Schedule		<u>Is the project running on schedule?</u>
Resources		<u>Are you low or have excess resources?</u>
Budget		<u>Are you on budget, under, or over?</u>
Quality		<u>Is the project's quality being jeopardized?</u>
Team Performance		<u>How is the team performing? Is someone in particular excelling?</u>
Roadblocks		<u>Potential risks & roadblocks</u>

Задание 4: Итоговый отчет

Сборка всех созданных артефактов в структурированный документ (PDF) или вики GitHub-репозитория для комплексного представления проделанной работы.

- 1 — Структура отчета
Титульный лист, введение, все артефакты по порядку
- 2 — Документация артефактов
Видение продукта, бэклог, протоколы, прототипы, диаграммы
- 3 — Технические материалы
Ссылка на репозиторий, тест-кейсы, скриншоты
- 4 — Заключение
Выводы о работе, трудности, полезные инструменты

Отчет должен демонстрировать полное понимание процесса создания артефактов проекта и их роли в жизненном цикле разработки ПО.