УТВЕРЖДЕН

10.04.2025-01 90 01-ЛУ

**Наименование проекта**

**Приложение «Будильник» (Alarm Clock) на платформе .NET/C#**

**02.12.2024-01 90 01**

**Листов 12**

2025

АННОТАЦИЯ

Данный документ содержит полное техническое задание на разработку программного продукта «Будильник». В документе описаны цели, требования, функциональные и эксплуатационные характеристики разрабатываемого приложения, этапы разработки, а также перечень документации, сопровождающей проект. Приложение предназначено для помощи пользователю в планировании дня посредством настройки и автоматического запуска сигналов-напоминаний с возможностями дополнительной настройки (повтор, выбор мелодии, настройка громкости и т.п.).

СОДЕРЖАНИЕ

[1. Введение 4](#_Toc457301676)

[2. Основания для разработки 5](#_Toc457301677)

[3. Назначение разработки 6](#_Toc457301678)

[4. Требования к программе или программному изделию 7](#_Toc457301679)

[4.1. Требования к функциональным характеристикам 7](#_Toc457301680)

[4.2. Требования к надежности 7](#_Toc457301681)

[4.3. Условия эксплуатации 7](#_Toc457301682)

[4.4. Требования к составу и параметрам технических средств 7](#_Toc457301683)

[4.5. Требования к информационной и программной совместимости 7](#_Toc457301684)

[4.6. Требования к маркировке и упаковке 7](#_Toc457301685)

[4.7. Требования к транспортированию и хранению 7](#_Toc457301686)

[4.8. Специальные требования 7](#_Toc457301687)

[5. Требования к программной документации 8](#_Toc457301688)

[6. Технико-экономические показатели 9](#_Toc457301689)

[7. Стадии и этапы разработки 10](#_Toc457301690)

[8. Порядок контроля и приемки 11](#_Toc457301691)

[Перечень принятых сокращений 12](#_Toc457301692)

1. Введение

Приложение «Будильник» – настольное и (при необходимости) мобильное приложение, реализованное на C#, предназначенное для своевременного извещения пользователя о наступлении заданного времени посредством звуковых сигналов, а также предоставляющее функциональные возможности для настройки повторений и индивидуализации сигналов.

Создать интуитивно понятное, надёжное и гибкое в настройке программное средство для ежедневного использования, способное работать на стандартном компьютере с ОС Windows (.NET Framework или .NET Core).

В условиях быстрого ритма жизни своевременное напоминание о встречах, задачах и событиях важно для повышения продуктивности. «Будильник» удовлетворяет данную потребность, предоставляя функциональные возможности для управления временем.

1. Основания для разработки

На современном рынке присутствует множество приложений с функциями будильника, однако большинство из них либо утрачивают актуальность из-за отсутствия гибкой настройки, либо требуют подключения к интернету для синхронизации.

Требования заказчика:

* Реализация на платформе C# с использованием современных паттернов проектирования
* Возможность работы в автономном режиме
* Высокая скорость отклика и надежность при выполнении задачи

Стандарты:

* Использование стандартов оформления кода и комментариев (XML-комментарии)
* Соблюдение требований безопасности при работе с пользовательскими данными
* Соответствие нормативным актам по защите информации

1. Назначение разработки

Планирования времени: Позволяет установить один или несколько будильников на определенное время, а также настроить повторения для будильника (ежедневно, по будням, выходные).

Управления уведомлениями: Пользователь может выбрать тип звукового сигнала, установить громкость и периодичность повторения сигнала, а также временной интервал для режима отсрочки (snooze).

Гибкой настройки: Приложение предоставляет расширенные настройки (темы, режим энергосбережения, интеграция с календарем) для более персонализированного использования.

1. Требования к программе или программному изделию
   1. Требования к функциональным характеристикам

Основные функции:

* Установка времени будильника с возможностью множественных установок
* Выбор звукового сигнала из набора предустановленных и возможность загрузки собственного файла
* Режим повторения будильника с настройками (ежедневно, выходные, конкретные дни недели)
* Функция "Отложить" (snooze) с настраиваемым интервалом времени
* Визуальное отображение активных будильников в интерфейсе
* Возможность выключения/активации будильника

Дополнительные функции:

* Настройка внешнего вида (темы, изменение цвета и размера шрифтов)
* Интеграция с календарем и напоминаниями через API (при наличии интернет-соединения)
* Логирование срабатываний, ошибок и действий пользователя для диагностики
  1. Требования к надежности
* Стабильность работы: Приложение должно демонстрировать высокую устойчивость к сбоям, обеспечивать автоматическое восстановление после ошибок.
* Обработка ошибок: Все критические ошибки должны записываться в лог-файл с указанием времени, причины и предложения по решению.
* Тестирование: Проведение модульного, интеграционного и системного тестирования.
  1. Требования к эксплуатации

Рабочая среда: Windows 7 и выше, рекомендации для использования Windows 10 или выше

Возможна кросс-платформенная реализация с использованием .NET Core (Linux, macOS)

Эксплуатация в автономном режиме: Приложение должно работать без подключения к сети интернет.

* 1. Требования к составу и параметрам технических средств

Минимальная конфигурация:

Процессор: 1 ГГц

Оперативная память: 512 МБ

Жесткий диск: не менее 100 МБ свободного места

Дисплей: разрешение не менее 1024×768 пикселей

Рекомендуемые параметры:

Двухъядерный процессор

ОЗУ: 2 ГБ

Наличие звуковой карты и динамиков

* 1. Требования к информационной и программной совместимости

Языки интерфейса: Русский, английский (с возможностью расширения)

Форматы данных: Сохранение настроек и логов в текстовых или XML/JSON файлах

Интеграция с другими программами: Возможность взаимодействия с календарем Windows, средствами планирования задач

* 1. Требования к маркировке и хранению

Маркировка продукта: Продукт должен быть снабжен уникальным идентификатором версии, названием и информацией о лицензировании.

Упаковка: В случае поставки как установочного пакета – требуется создать графический дизайн установщика и сопроводительную документацию (инструкция, лицензионное соглашение).

* 1. Специальные требования

Транспортировка: Электронный носитель (интернет-распространение или физический носитель) с защитой от несанкционированного доступа.

Хранение: Документация и исходники должны храниться в системе контроля версий с обеспечением резервного копирования.

1. Требования к программной документации

**Обязательные документы:**

* 1. Техническое задание (настоящий документ)
  2. Текст программы с подробными комментариями
  3. Описание программы (логическая структура, схема алгоритма, принципы работы)
  4. Пояснительная записка (описание алгоритмов, обоснование архитектурных решений)
  5. Эксплуатационные документы, включающие:
     + Описание применения
     + Руководство системного программиста
     + Руководство программиста
     + Руководство оператора
     + Описание языка (при необходимости)
     + Руководство по техническому обслуживанию

**Общий формат:** Документы должны быть оформлены в едином стиле, с указанием версии, даты утверждения и подписей ответственных лиц.

1. Технико-экономические показатели

Для определения затрат на разработку используется методика «Function Point» (FP), которая позволяет оценить сложность проекта по количеству функциональных элементов и вычислить трудоёмкость.

1. Анализ функциональных точек

Примерная оценка функциональных точек для базового приложения «Будильник» может включать следующие компоненты:

* **Ввод данных:** Ввод времени, настройка повторов, выбор звука – 10 FP
* **Обработка и логика:** Основной алгоритм проверки времени, обработка функции отсрочки (snooze) – 15 FP
* **Вывод/интерфейс:** Отображение списка будильников, уведомления – 10 FP
* **Дополнительный функционал:** Логирование событий, настройка темы, интеграция с календарём (при наличии) – 5–10 FP

Общий объём FP можно взять в диапазоне 40–45 FP для версии с базовым функционалом. Для расчёта трудозатрат применяется следующая модель:

Трудоёмкость (челов./час)=FP×KТрудоёмкость (челов./час)=FP×K

где коэффициент KK зависит от сложности проекта (как правило, для средних по сложности систем принимается значение K≈8−10K≈8−10 часов на FP). В нашем примере возьмём K=9K=9 часов.  
45 FP × 9 ч/FP = 405 человеко-часов

Проект выполняется командой из нескольких специалистов.

| **Специалист** | **Объем работы (часы)** | **Ставка (руб./час)** | **Стоимость (руб.)** |
| --- | --- | --- | --- |
| **Разработчик (C#)** | 250 | 800 | 250 × 800 = 200 000 |
| **Тестировщик** | 80 | 600 | 80 × 600 = 48 000 |
| **UI/UX дизайнер** | 40 | 700 | 40 × 700 = 28 000 |
| **Технический писатель** | 35 | 500 | 35 × 500 = 17 500 |
| **Менеджер проекта** | 30 | 900 | 30 × 900 = 27 000 |
| **Итого** | – | – | ~320 500 руб. |

Кроме прямых затрат на оплату труда, обычно включают накладные расходы (офисные расходы, оборудование, лицензии, средства поддержки и пр.). Примем коэффициент накладных расходов около 15–20% от прямых затрат.

Расчёт накладных расходов:

320 500 руб. × 20% = 64 100 руб.

Общая смета затрат на разработку:

320 500 руб. + 64 100 руб. = 384 600 руб. (округленно ~385 тыс. руб.)

7. Стадии и этапы разработки

**Анализ требований:**

* Сбор и анализ пожеланий заказчика
* Изучение аналогов и определение области применения

**Проектирование:**

* Разработка архитектуры приложения
* Проектирование интерфейса пользователя
* Подготовка схем алгоритмов работы

**Разработка:**

* Написание исходного кода на языке C#
* Разработка модулей и интеграционных тестов

**Тестирование:**

* Модульное тестирование каждого компонента
* Интеграционное тестирование системы
* Системное тестирование в реальных условиях

**Ввод в эксплуатацию:**

* Подготовка документации и установочного пакета
* Проведение обучения для оператора и персонала

**Сопровождение:**

* Обеспечение технической поддержки
* Выпуск обновлений и исправлений ошибок

8. Порядок контроля и приемки

**Контроль качества:**

* Проведение ревью кода и документации
* Организация испытательного периода с участием тестовой группы пользователей

**Критерии приемки:**

* Соответствие функциональным требованиям
* Отсутствие критических ошибок и сбоев
* Наличие полной документации и инструкций по эксплуатации

**Порядок приемки:**

* Проведение демонстрационных тестов
* Согласование отчета о тестировании с представителями заказчика
* Окончательная подписная приемка и утверждение документации

Перечень принятых сокращений

* ПК – Персональный компьютер
* UI – User Interface (Пользовательский интерфейс)
* API – Application Programming Interface
* XML – Extensible Markup Language
* JSON – JavaScript Object Notation
* ОЗУ – Оперативное запоминающее устройство
* SDK – Software Development Kit